8-1-2012

Modeling Daily Power Demand in Southern Kentucky: A Single Household Approach

Craig M. Dickson
Western Kentucky University, craig.dickson@topper.wku.edu

MODELING DAILY POWER DEMAND IN SOUTHERN KENTUCKY: A SINGLE HOUSEHOLD APPROACH

A Thesis
Presented To
The Faculty of the Department of Mathematics
Western Kentucky University
Bowling Green, Kentucky

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science

By
Craig M. Dickson

August 2012

MODELING HOUSEHOLD HOURLY ELECTRICITY DEMAND IN SOUTHERN KENTUCKY: A SINGLE HOUSEHOLD APPROACH

Date Recommended July 13, 2012

Jonathan Quiton

Dr. Jonathan Quiton, Director of Thesis

Mark P. Robinson

Dr. Mark Robinson

Melanie A. A

Dr. Melanie Autin

8/23/12

Dean, Graduate Studies and Research Date

ACKNOWLEDGMENTS

I can say honestly that Dr. Jonathan Quiton might be the most patient person I have ever met. The impact he has had on my life and my professional development cannot be overstated. This thesis is a product of his guidance and understanding as a professional and a mentor. It is not possible to return his gratitude, only to help others as he has helped me.

I gratefully acknowledge Dr. Mark Robinson and Dr. Melanie Autin for the many hours I spent in their offices coming to grips with the knowledge required to write this thesis. They were my professors and were good company on mornings before homework was due. It goes without saying that they look forward to seeing me graduate.

I acknowledge research support from Kentucky Science and Engineering Foundation (KSEF-2013-RDE-012), Western Kentucky University Junior Faculty Scholarship (No. 223149), and computing support from the Kentucky NSF- EPSCoR Research Startup Fund (RSF-031-06). In addition, I would like to thank Mr. Josh Francis, Mr. Daniel Mooney, and Engr. William Ray, CEO of the Glasgow Electric Plant Board. The data they provided made this study possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

MODELING HOUSEHOLD HOURLY ELECTRICITY DEMAND IN SOUTHERN KENTUCKY: A SINGLE HOUSEHOLD APPROACH

Craig M. Dickson August 2012 67 Pages

Directed by: Dr. Jonathan Quiton, Dr. Mark Robinson and Dr. Melanie Autin

Department of Mathematics Western Kentucky University

In this study, we use a nonparametric technique, locally weighted robust least squares regression (LOESS), to forecast a 24 hour demand profile at the household level and compare it to existing aggregate demand models discussed in literature. Of these aggregate demand models, a quadratic autoregressive model was selected to be used as a basis for comparison with the LOESS forecasts. It was our goal to automate the forecasting process by using the goodness of fit metric, AICC1, for smoothing parameter selection. The statistical workflow was executed using SAS and data was provided by the Glasgow Electric Plant Board of Barren County, Kentucky. Results show that LOESS outperformed the autoregressive model in roughly 80% of all cases and that using LOESS alone or as part of an ensemble model is a feasible approach to automating future household demand profile for the purpose of generating different levels of power demand profile aggregation as needed by Glasgow Electric Plant Board.

INTRODUCTION

As the demand for energy increases throughout the world, there is a growing sense of urgency to monitor and consume it more efficiently. The term "smart grid" is often used to describe a system which meets those needs ([8]). While it serves as a compact description for what is now an established marketing trend (i.e. smart phones, smart cars, smart appliances, etc.), it is simply not one thing or one industry. It is a global movement to harness the data available to us through technological innovation to repair the inefficiencies, and our inefficiencies as consumers, in the power grid. The focus of this research as it pertains to the smart grid initiative is the latter; the ability to inform the power company and the consumer of their usage ahead of time is an invaluable tool for altering consumer behavior and limiting demand spikes.

The Glasgow Electric Plant Board (EPB) in Southern Kentucky has already begun to outfit a small portion of their customers with internet protocol based smart meters, which have the ability to capture and relay usage data on a real time basis. This data can be stored and analyzed to give meaningful feedback to these customers. In this study, we process Glasgow's data and construct a model to forecast demand by the hour for any household in the grid. This will allow us to generate a 24-hour power demand profile for the households. Our primary motivation is to predict critical usage times, or periods of extreme load. With this information, Glasgow hopes to see a decrease in demand spikes by incentivizing individuals to consume more uniformly throughout the day. The challenge in doing this is different from a standard global consumption model where demand is considered for an entire region. Such models required testing on individual households to evaluate their effectiveness.

1.1 Objectives

A global predictive model considers every data point simultaneously to estimate the parameters of the model. A study of the data provided by the Florida Power Corporation is discussed by Mendenhall and Sincich and shows that parametric regression-autoregression techniques are effective in predicting demand for aggregate consumption data [9]. A priority of this study was to explore the predictive power of the models presented using the technique at the household level. One model, a quadratic autoregressive model, was selected amongst four candidate models that are presented in chapter 2.

In contrast to a global model, a local predictive model focuses on portions of the data, and uses a series of models for each portion with their own parameters. A nonparametric technique for local predictive modeling is locally weighted robust least squares regression (LOESS). In effect, there is not an assumed relationship between the response and explanatory variables by the researcher when the model is applied. LOESS, is the candidate nonparametric technique which is competing against the parametric technique, regression-autoregression. It is our goal to explain in detail the mathematical construction of LOESS, how and what it is designed to do, and why we chose it for modeling power demand in this study.

The two models are applied to data collected by the Glasgow EPB. We consider the effectiveness of forecasts made by both models by comparing root mean cross validation error (RMCVE), discussed in chapter 3. The purpose of this comparison is to create a decision rule for when to use one model over the other. The results are aggregated by the household location I.D. and by hour of the day.

The final product of this research is the creation of an automated system for producing a graphical 24-hour demand profile for any participant in Glasgow's program. These profiles can be used to relay predicted usage information to the customers to help prevent power demand spikes in the grid. A sample profile is presented at the end of chapter three.

METHOD

In this chapter we discuss selected aggregate models found in literature and present a mathematical framework for locally weighted robust least squares regression (LOESS).

2.1 Aggregate Demand Models

The Florida Power Corporation provided aggregate data from November 1, 1982 to October 31, 1983 to develop a forecasting model for daily peak electricity demand. Let the demand on day t be denoted as y_t. Researchers presented three piece-wise models which were then evaluated based on their complexity and goodness of fit to the data [9].

2.1.1 Model 1

Let $i = 1, 2, \cdots, n$ be the index for the i^{th} day in the data set and n be the sample size. Model 1 is then described as

$$y_i = \beta_0 + \beta_1(x_{1i} - 59)x_{2i} + \beta_2(x_{1i} - 78)x_{3i} + \beta_3 x_{4i} + \beta_4 x_{5i} + \varepsilon_i, \tag{2.1}$$

where

$$y_i = \text{Response variable for day } i,$$

$$x_{1i} = \text{Coincident temperature for day } i,$$

$$x_{2i} = \begin{cases} 1 & \text{if } x_{1i} < 59 \\ 0 & \text{otherwise,} \end{cases} \qquad x_{3i} = \begin{cases} 1 & \text{if } x_{1i} > 78 \\ 0 & \text{otherwise,} \end{cases}$$

$$x_{4i} = \begin{cases} 1 & \text{if Saturday} \\ 0 & \text{otherwise,} \end{cases} \qquad x_{5i} = \begin{cases} 1 & \text{if Sunday/holiday} \\ 0 & \text{otherwise.} \end{cases}$$

$\varepsilon_i =$ residual for day i

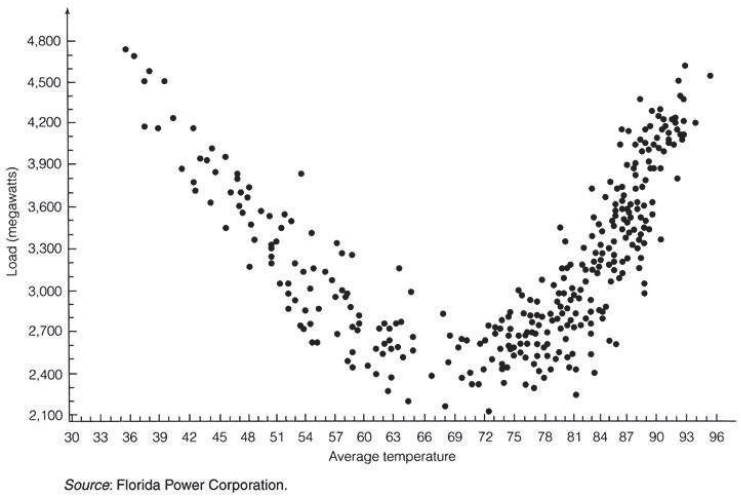

Figure 2.1: A U-shaped scatter plot of consumption within the FLPC grid

The model illustrates the relationship between coincident temperature and daily peak demand using three line segments which correspond to weather sensitive time periods. Researchers were able to determine the values of $78°$ and $59°$ by inspecting Figure 2.1 [9]. They are expecting that at certain times of the year the factors that drive demand are independent of temperature. In such cases x_{2t} and x_{3t} will be evaluated as zero, and the only contributing deterministic factors will be the day of the week and/or holidays.

Model 1 is a multiple regression model that relies on the standard regression assumptions, and rarely are these assumptions satisfied, especially in the context of time series modeling.

Standard Regression Assumptions:

Let $\{\varepsilon_i : i = 1, 2, \cdots, n\}$ be a sequence of error terms associated with Model 1 and $\boldsymbol{\varepsilon}$ be an $n \times 1$ vector representation of $\{\varepsilon_i\}$. Let $\mathbf{Y}$ be an $n \times 1$ vector of responses and $\mathbf{X}$ be a $n \times p$ vector of predictors (for Model 1, $p = 5$). Then, our standard regression assumptions are

(a) $\{\varepsilon_i\}$ are independent and normally distributed error terms.

(b) $\mathbf{E}[\boldsymbol{\varepsilon}] = \mathbf{0}$ is an $n \times 1$ vector of zeroes.

(c) Homoscedasticity: $\mathbf{V}[\boldsymbol{\varepsilon}] = \sigma^2\mathbf{I}$ an $n \times n$ symmetric matrix

(d) Predictors are linearly independent. The implication of this is that $\mathbf{X}$, the matrix of has a full column rank such that $\mathbf{X}^T\mathbf{X}$ is nonsingular. For Model 1, $\mathbf{X}^T\mathbf{X}$ is a 5×5 nonsingular symmetric matrix.

These assumptions are often disrupted by the nature of the data and our choice of models. Autocorrelation, lack of fit, heteroscedasticity, and multicollinearity are terms for violations of assumptions (a), (b), (c) and (d) respectively. Remedial measures can be taken to correct for such things. Graphical methods are a common way to detect violations. For instance, in theory the residuals, ε_i, of the regression model should be scattered randomly about the line $y = 0$. Any patterns could indicate an unstable variance problem, a lack of fit, or serial correlation. Once this is seen, an analytical method can be invoked to verify the violation. In the case of first-order serial correlation the Durbin-Watson test is such a method [9]. In this study, a SAS printout shows that first-order serial correlation was detected and a regression-autoregression model was explored in model 2.

2.1.2 Model 2

Model 2,

$$y_t = \beta_0 + \beta_1(x_{1t} - 59)x_{2t} + \beta_2(x_{1t} - 78)x_{3t} + \beta_3 x_{4t} + \beta_4 x_{5t} + R_t, \tag{2.2}$$

where

$$R_t = \phi_1(R_{t-1}) + \varepsilon_t,$$

is a regression-autoregression model that corrects for first-order serial correlation in the error term. A first-order model is appropriate when the correlation between residuals diminishes as the gaps between them widen. It states that a residual at time t is a function of its uncorrelated residual and the previous residual which is scaled by a factor of ϕ_1. The null hypothesis of the Durbin-Watson test assumes $\phi_1 = 0$ (no autocorrelation) and, in this case, that the alternative is $\phi_1 > 0$. This can be extended using the Generalized Durbin-Watson Test to j lags [3]. In model 3 this comes into play.

2.1.3 Model 3

Model 3,

$$y_t = \beta_0 + \beta_1(x_{1t} - 59)x_{2t} + \beta_2(x_{1t} - 78)x_{3t} + \beta_3 x_{4t} + \beta_4 x_{5t} + R_t, \tag{2.3}$$

where

$$R_t = \phi_1(R_{t-1}) + \phi_2(R_{t-2}) + \phi_5(R_{t-5}) + \phi_7(R_{t-7}) + \varepsilon_t,$$

extends (2.2) from first-order to seventh-order. Perhaps the researchers believed that autocorrelation was present up to one week. Most statistical software packages are able to test for significant lags up to a number set by the researcher. A SAS printout provided in the study confirms high correlation at $t-1$, $t-2$, $t-5$, and $t-7$, meaning these

residuals will influence the outcome of R_t [9].

2.1.4 Model 4

Models one through three attack the problem of forecasting demand with a piece-wise model. However, Figure 2.1 can be described as having a parabolic shape. While mentioning a quadratic least-squares model,

$$y_t = \beta_0 + \beta_1 {x_{1t}}^2 + \beta_2 x_{1t} + \beta_3 x_{4t} + \beta_4 x_{5t} + \varepsilon_t, \tag{2.4}$$

researchers in the study dismissed the possibility for two reasons: one, due to the symmetric shape of the model, it would not allow for independent estimates of the winter and summer peak demand-temperature relationship, and two, based on their theory that demand is independent of temperature at certain times of the year, the parabolic shape would underestimate demand near the center of mild temperature ranges and over estimate it near the outskirts around $59°$ and $78°$. This can be seen in Figure 2.2 [9].

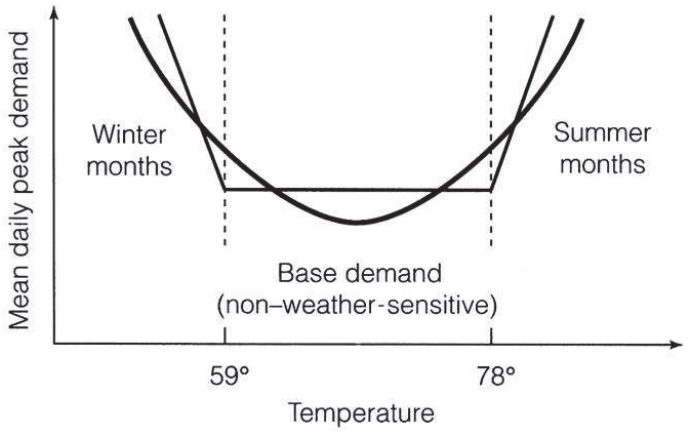

Figure 2.2: A comparison between the proposed piece-wise and quadratic models [9]

These problems are specific to the data presented by the Florida Power Corporation. It will be shown later in the chapter that hourly demand profiles seldom follow the "59/78" model. Needless to say, "eye-balling" the proper temperature for each hour will not be an

option if we hope to automate the process. This is why we choose model four as our candidate parametric model, which is discussed more in the results section.

2.2 Mathematical Framework for Least Squares Regression

It is the goal of this section to describe the foundation for global least squares regression, which will then be extended to local regression in the context of LOESS. Since our model uses just one predictor, we will proceed with the derivation as such.

2.2.1 Ordinary Least Squares (OLS) Regression

Consider

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i \qquad \text{for } i = 1, 2, ..., n \tag{2.5}$$

as the relationship between two variables X and Y. Let β_0 be the intercept term, β_1 reflect the slope of the line, and ε_i be the error, or vertical distance between the actual point (X_i, Y_i) and the line. This is the setting for a simple linear regression model. Equation (2.5) implies

$$\begin{aligned} Y_1 &= \beta_0 + \beta_1 X_1 + \varepsilon_1 \\ Y_2 &= \beta_0 + \beta_1 X_2 + \varepsilon_2 \\ &\vdots \\ Y_n &= \beta_0 + \beta_1 X_n + \varepsilon_n, \end{aligned}$$

which can be represented using matrices in the following manner:

$$\mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_n \end{bmatrix} \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_0 \\ \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix}.$$

where $\mathbf{Y}$ is an $n \times 1$ vector of responses, $\mathbf{X}$ is an $n \times 2$ vector of predictors (the first column being a column of one's), and $\boldsymbol{\varepsilon}$ is an $n \times 1$ vector of error terms. This allows us to express Equation 2.5 as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{2.6}$$

and the error term as

$$\boldsymbol{\varepsilon} = \mathbf{Y} - \mathbf{X}\boldsymbol{\beta}. \tag{2.7}$$

In practice, $\boldsymbol{\beta}$ is unknown and the goal of the ordinary least squares approach is to find an estimator for $\boldsymbol{\beta}$ that will minimize the sum of squares error,

$$SSE = \boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon} = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}). \tag{2.8}$$

Distributing the terms accordingly we have the following results:

$$SSE = \mathbf{Y}^T\mathbf{Y} - \mathbf{Y}^T\mathbf{X}\boldsymbol{\beta} - \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{Y} + \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta}. \tag{2.9}$$

$$= \mathbf{Y}^T\mathbf{Y} - 2\boldsymbol{\beta}^T\mathbf{X}^T\mathbf{Y} + \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta}. \tag{2.10}$$

To minimize (2.10) we must take the partial derivative with respect to $\boldsymbol{\beta}$, giving

$$\begin{aligned} \frac{\partial SSE}{\partial \boldsymbol{\beta}} &= -2\mathbf{X}^T\mathbf{Y} + (\mathbf{X}^T\mathbf{X})\boldsymbol{\beta} + (\mathbf{X}^T\mathbf{X})\boldsymbol{\beta} \\ &= -2\mathbf{X}^T\mathbf{Y} + 2(\mathbf{X}^T\mathbf{X})\boldsymbol{\beta}. \end{aligned} \tag{2.11}$$

Finally, set (2.11) equal to zero we obtain the estimator for $\boldsymbol{\beta}$, giving

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y}). \tag{2.12}$$

Equation 2.12 is the ordinary least squares (OLS) estimator for the regression parameters. This formula holds true for any p number of predictors.

As discussed previously in the chapter, the assumptions of regression models are often disrupted. LOESS compensates for violations using weighted regression techniques.

2.3 Weighted Least Squares Regression

We continue to build on the concepts from OLS regression by introducing a weight into the regression model. Weights can be used to modify an observation with respect to its squared error in the following manner:

$$WSSE = \sum_{i=1}^{n} w_i {\varepsilon_i}^2, \tag{2.13}$$

where w_i is an assigned weight for the i^{th} residual. In matrix form, let

$$\mathbf{W} = \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{bmatrix} \tag{2.14}$$

be a diagonal matrix of weights. Then

$$WSSE = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T \mathbf{W} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) \tag{2.15}$$

is to be minimized in the same manner as (2.8). The final estimator, $\widehat{\boldsymbol{\beta}}$ is derived in a similar fashion to (2.12),

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} (\mathbf{X}^T \mathbf{W} \mathbf{Y}). \tag{2.16}$$

If we let $\mathbf{D}$ to be a diagonal matrix of the square-roots of the weights found in $\mathbf{W}$ such that $\mathbf{W} = \mathbf{D}\mathbf{D}$ and use the transformed $\mathbf{X}$ and $\mathbf{Y}$ as $\mathbf{X}^* = \mathbf{D}\mathbf{X}$ and $\mathbf{Y}^* = \mathbf{D}\mathbf{Y}$, respectively, we obtain a similar result (2.12) under the transformed variables,

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}^{*T} \mathbf{X}^*)^{-1} (\mathbf{X}^{*T} \mathbf{Y}^*). \tag{2.17}$$

There are several reasons for wanting to modify the residuals of an observation. Mainly, it stems from the fact that residuals are closely tied to the assumptions made by the regression model. Mendenhall and Sincich [9] provide us with a few:

(a) Stabilizing the variance of ε to satisfy the standard regression assumption of homoscedasticity.

(b) Limiting the influence of outlying observations on the regression analysis.

(c) Giving greater weight to more recent observations in time series analysis.

The weight function is largely dependent on the application. When it comes to outlying observations, they tend to skew the regression line toward the deviant point, effectively causing it to over or under predict. Keep in mind that the least squares method is trying to minimize the sum of squared errors and not the absolute errors. This can affect how the method calculates the regression parameters because it sees outlying observations as being overwhelmingly influential. For example, imagine a linear regression has just been

calculated for three points $(1, 1)$, $(2, 2)$, and $(3, 3)$. The regression line is simply $Y = X$, which is a perfect linear regression equation. Suppose a fourth point is inserted at $(4, 12)$. With the addition of the fourth point, our line of best fit becomes $y = -4 + 3.4X$, changing our original slope and y-intercept drastically, which means the fourth point is both an outlier and an influential point. Now consider the original three points, but replace $(2, 2)$ with $(2, 12)$. The new regression line would be $y = 1/3 + X$. In this case, $(2, 12)$ may be an outlying point but is not as influential as $(4, 12)$.

As for (c), we discuss in the next section how Cleveland [1] uses weights to exclude far off points and which weight functions he uses to define LOESS.

2.4 Locally Weighted Robust Least Squares Regression

LOESS can be identified as a method of fitting a series of regression lines to the data by forming neighborhoods centered around a smoothing point, (x_i, y_i), in a manner which can minimize the effects of deviant points using weights [1]. In this section we precisely define the procedure according to Cleveland (with a few minor changes in notation) by exploring the weight functions, the order in which they are to be used, and how we can form neighborhoods given a smoothing parameter.

2.4.1 The Tricube Weight Function

Let C be a weight function with the following properties:

(a) $C(x) > 0$ for $|x| < 1$;

(b) $C(-x) = C(x)$;

(c) $C(x)$ is a nonincreasing function for $x \geq 0$;

(d) $C(x) = 0$ for $|x| \geq 1$.

For LOESS, the purpose of C is to limit the influence of points whose abscissas lie further from x_i and to provide starting weights for future iterations. In his original paper [1], Cleveland defined the tricube weight function as

$$C(x) = \begin{cases} (1-|x|^3)^3 & \text{for } |x| < 1. \\ 0 & \text{for } |x| \geq 1 \end{cases} \tag{2.18}$$

An alternative form would be $C(x) = (1-|x|^3)^3 I\left(|x| < 1\right)$, where $I(\cdot)$ is an indicator function. The "tricube" weight function, as Cleveland states, "was chosen since...it enhances a chi-squared distributional approximation of an estimate of the error variance [1]." It also satisfies the necessary properties for a weight function stated previously in this section. This is, however, a general form for producing weights, and we modify the tricube to suit our needs.

Assignment of the weight function dynamically changes as we predict using one x-value at a time. For example, let x_i be our center point and let q be the smoothing parameter such that x_q is the q^{th} order statistic relative to x_i. Then, we assign the weight

$$c_k(x_i) = C\left(\frac{d_k}{d_q}\right), \tag{2.19}$$

where $d_k = |x_k - x_i|$ and $d_q = |x_i - x_q|$, for $k = 1, ..., n$. We then construct the weight matrix as

$$\mathbf{W} = \begin{bmatrix} c_1(x_i) & 0 & \cdots & 0 \\ 0 & c_2(x_i) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_n(x_i) \end{bmatrix}. \tag{2.20}$$

The initial fitted value, $\widehat{y}_i$, at x_i is the result of fitting a d^{th} degree polynomial using weighted least squares with weights $\{c_k(x_i) : k = 1, 2, \cdots, n\}$. We repeat the process by

letting another point x_i be the center point. The whole process of iteratively fitting weighted regression using is referred to as locally weighted regression [1].

To illustrate, consider the following numerical example. Let $X = \{10, 11, 13, 14, 18\}$ and $Y = \{7, 19, 6, 6.2, 6.1\}$ and let the matrix $\mathbf{X} = [\mathbf{1}X]$ where $\mathbf{1}$ is a vector of 1's. Let $x_i = x_3 = 13$ be the center point and let $x_q = x_4 = 10$ be the farthest point from the center with respect to the x-axis. Consequently, $d_q = |10 - 13| = 3$, and the corresponding sequence of absolute distances d_k would be:

$$\begin{aligned} d_1 &= |10 - 13| = 3, \\ d_2 &= |11 - 13| = 2, \\ d_3 &= |13 - 13| = 0, \\ d_4 &= |14 - 13| = 1, \\ d_5 &= |18 - 13| = 5, \end{aligned}$$

leading to our sequence of weights

$$\begin{aligned} c_1(x_3) &= \left(1 - \left\{\frac{3}{3}\right\}^3\right)^3 I\left(\left|\frac{3}{3}\right| < 1\right) = 0, \\ c_2(x_3) &= \left(1 - \left\{\frac{2}{3}\right\}^3\right)^3 I\left(\left|\frac{2}{3}\right| < 1\right) = 0.35, \\ c_3(x_3) &= \left(1 - \left\{\frac{0}{3}\right\}^3\right)^3 I\left(\left|\frac{0}{3}\right| < 1\right) = 1, \\ c_4(x_3) &= \left(1 - \left\{\frac{1}{3}\right\}^3\right)^3 I\left(\left|\frac{1}{3}\right| < 1\right) = 0.89, \\ c_5(x_3) &= \left(1 - \left\{\frac{5}{3}\right\}^3\right)^3 I\left(\left|\frac{5}{3}\right| < 1\right) = 0. \end{aligned}$$

We let $\mathbf{W} = \text{diag}\{c_k(x_3) : k = 1, 2, \cdots, 5\}$ and using (2.16) we obtain our local linear regression

$$\widehat{Y} = 61.79 - 4.10X, \tag{2.21}$$

and the smoothed point $(x_3, \widehat{y_3}) = (13, 8.46)$.

In computing software like R or SAS, the smoothing parameter is expressed in terms of the proportion of the whole data set

$$f = \frac{q}{n} \qquad \text{for } 1 \leq q \leq n, \tag{2.22}$$

where $q = fn$ is rounded to the nearest integer. The use of f instead of q enables us to maintain the proportion of neighborhood points relative to the sample size. As the term implies, the smoothing parameter, f, determines the smoothness of the fitted curve. For example, Figure 2.3 shows the resulting LOESS fit to the same data using two different smoothing parameters. The graph shows that as f becomes larger, the smoother the line. When $f = 1$ then LOESS becomes a global model with only one set of weights to use.

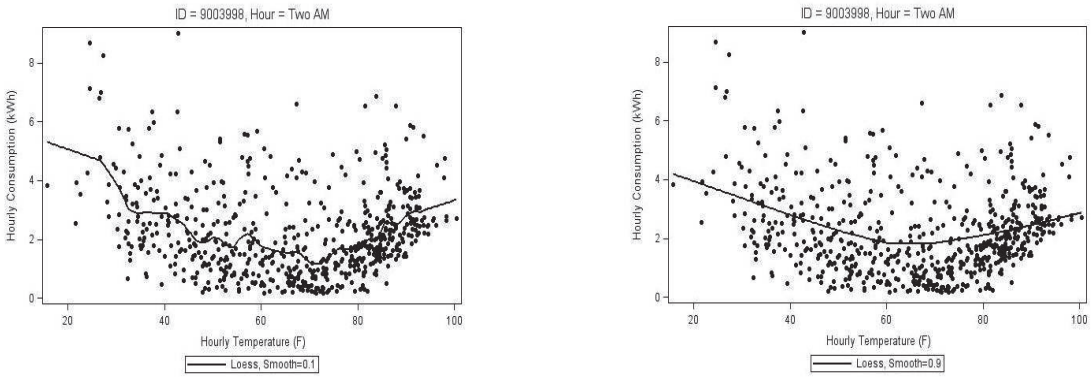

Figure 2.3: Contrasted smoothing parameters, for f=0.1 (left) and f=0.9 (right).

2.4.2 Iteratively Reweighted Least Squares (IRLS) Robust Regression

The last piece, IRLS, helps deal with outlying observations which might pull the regression line toward the deviant point. Outliers can also cause a departure from normality by creating a heavy tail in the error distribution, which violates a regression assumption. The following steps outline the procedure [7]:

1. Choose a weight function (e.g., the tricube function).
2. Iteratively obtain local weights by using x_i as the center.
3. For each x_i and its corresponding local weights, we obtain the fitted value $\widehat{y}_i$, and consequently, obtain our residuals $e_i = y_i - \widehat{y}_i$.
4. Collect all the residuals and obtain a sequence of weights $\{s_i : i = 1, 2, \cdots, n\}$, which serves as a correction factor to the local weights as we redo the local regression fitting process.

Cleveland recommends the bisquare function as the weight function for $\{s_i\}$ [1], which is

$$S(x) = \begin{cases} (1 - x^2)^2 & \text{for } |x| < 1 \\ 0 & \text{for } |x| \geq 1, \end{cases} \tag{2.23}$$

where $x = \frac{|e_k|}{6m}$ is a standardized residual and $m = \text{median } \{|e_k| : k = 1, 2, \cdots, n\}$. The bisquare is subject to sensitivities in starting values and generally is not the first weight function to be used. In practice, the researcher chooses another weight function to form an initial regression and then the bisquare is used in successive iterations [7]. In our case, the starting weights are given by the tricube. Since Cleveland's first publication, more advanced scaling methods have been developed for weighted least squares to provide an unbiased estimate of σ, given independent observations from a normal distribution. See Kutner et. al for more information [7].

To summarize, here is the precise instruction on how to execute LOESS with respect to this research:

1. For each x_i obtain the local parameter estimate, $\widehat{\boldsymbol{\beta}}$, of the polynomial regression of degree d using weighted least squares given as the following:

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{W}\mathbf{Y}),$$

   such that we obtain $\widehat{y_i} = \mathbf{X}_i\widehat{\boldsymbol{\beta}}$ where $\mathbf{X}_i$ is the i^{th} row of the matrix $\mathbf{X}$.

2. Collect all the residuals $\{e_i = y_i - \widehat{y_i} : i = 1, 2, \cdots, n\}$, standardize them by dividing the vector by $6m$, and obtain the vector of correction factors s_i based on the bisquare function.

3. Redo the local fitting in step 1, but this time multiply each weight by the correction factor; i.e, weight $s_k c_k(x_i)$, which is now used to obtain a new fitted value, $\tilde{y}_i$.

4. We then collect all the residuals $\{d_i = y_i - \tilde{y}_i\}$, and standardize them by dividing by $6m$, where m is the median of the absolute values of d_i. From the standardized residuals, we obtain a new correction factor, r_k.

5. We then redo the local fitting in step 1, this time applying two correction factors to the original weight factor $r_k s_k c_k(x_i)$, which is used to estimate y_i. The final $\widehat{y_i}$ are the robust locally weighted regression fitted values.

LOESS is a derivative of IRLS Robust Regression, but it is used at the local level. It also differs in that it combines two weight functions to produce fitted values instead of one. Local fitting of polynomials had been in place for quite some time before LOESS was developed, but the fundamental advancement made by Cleveland was the introduction of a method that was more accommodating to unevenly spaced x_i [1]. In effect, LOESS is a generalization of all local regression models where the degree of the polynomial, the

number of explanatory variables, and the spacing of those variables are allowed and decided by the researcher. This means that any nonlinear curve, such as those in this power demand study, can be approximated by a series of simple (but not limited to) linear models.

As we saw in Figure 2.3, the smoothing parameter f plays a huge factor in the curve fitting process, so it is important that we find a data driven method to find the optimal value of f. The common strategy is to use a goodness of fit metric, such as the sum of squares residual (SSE) or the coefficient of determination (R squared), such that we pick the value of f that minimizes SSE or maximizes R-squared. Another measure is the Akaike Information Criterion, or AIC, and later variants AIC_{C0} and AIC_{C1}, which takes into consideration model complexity in addition to minimizing the discrepancy function.

2.5 AIC_{C1}

AIC, like many goodness of fit statistics, tries to quantify the information lost by using a model to describe reality. It does so by identifying the tradeoff between model complexity and accuracy. It was originally designed to be used for parametric models as an approximately unbiased estimate of the expected Kullback-Leibler information. AIC_{C0} and AIC_{C1} are similar to AIC, but are used in the context of nonparametric regression. These two statistics are essentially the same, with AIC_{C1} being a computationally friendly approximation to AIC_{C0} [4].

To understand, we must show how to find the Kullback-Leibler (KL) discrepancy function and how it relates to AIC_{C0}. To begin, we define two models:

$$\mathbf{y} = \mathbf{m} + \boldsymbol{\varepsilon}, \qquad \text{where } \boldsymbol{\varepsilon} \backsim N(0, {\sigma_0}^2 \mathbf{I_n}) \tag{2.24}$$

and

$$\mathbf{y} = \boldsymbol{\mu} + \boldsymbol{\eta}, \qquad \text{where } \boldsymbol{\eta} \backsim N(0, \sigma^2 \mathbf{I_n}). \tag{2.25}$$

where $\mathbf{m}$ can be thought of as the "true model" for the data, and $\boldsymbol{\mu}$ is the researcher's model [4]. Suppose $f(\mathbf{y})$ is the likelihood function for (2.24); then

$$d(\mu, \sigma^2) = E_0[-2 \log f_1(\mathbf{y})] \tag{2.26}$$

is defined as the KL discrepancy function [4]. We start with the assumption that $\mathbf{y}$ is normally distributed; then

$$\begin{aligned} d(\mu, \sigma^2) &= E_0[-2 \log f_1(\mathbf{y})] \\ &= E_0[-2 \log(\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{-\frac{(y_i - \mu_i)^2}{2\sigma^2}\})] \\ &= E_0[-2 \log((2\pi\sigma^2)^{-\frac{1}{2}n} \exp\{-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \mu_i)^2\})], \end{aligned} \tag{2.27}$$

where E_0 is the expectation with respect to the true model. This further simplifies to

$$d(\mu, \sigma^2) = n \log(2\pi\sigma^2) + \frac{1}{\sigma^2} \sum_{i=1}^{n} E_0[(y_i - \mu_i)^2]. \tag{2.28}$$

We add and subtract m_i and distribute the expectation and use the facts that $E_0(y_i - m_i)(m_i - \mu_i) = 0$ and $E_0(y_i - m_i)^2 = \sigma_0^2$, giving

$$\begin{aligned} d(\mu, \sigma^2) &= n \log(2\pi\sigma^2) + \frac{1}{\sigma^2} \sum_{i=1}^{n} \left[E_0(y_i - m_i)^2 + E_0(m_i - \mu_i)^2\right] \\ &= n \log(2\pi\sigma^2) + \frac{1}{\sigma^2} \left[n{\sigma_0}^2 + \sum_{i=1}^{n} E_0(m_i - \mu_i)^2\right]. \end{aligned} \tag{2.29}$$

It will be useful here to revert back to matrix notation, giving

$$d(\boldsymbol{\mu}, \sigma^2) = n \log(2\pi\sigma^2) + \frac{n{\sigma_0}^2}{\sigma^2} + \frac{(\mathbf{m} - \boldsymbol{\mu})^T (\mathbf{m} - \boldsymbol{\mu})}{\sigma^2}. \tag{2.30}$$

Now let $\boldsymbol{\mu} = \widehat{\mathbf{m}}$ be an estimate of $\mathbf{m}$ and $\sigma^2 = \widehat{\sigma}^2$ be an estimate of the variance. Then,

$$d(\widehat{\mathbf{m}}, \widehat{\sigma}^2) = n \log(2\pi\widehat{\sigma}^2) + \frac{n{\sigma_0}^2}{\widehat{\sigma}^2} + \frac{(\mathbf{m} - \widehat{\mathbf{m}})^T(\mathbf{m} - \widehat{\mathbf{m}})}{\widehat{\sigma}^2}. \tag{2.31}$$

As Hurvich, Simonoff, and Tsai state,"A reasonable criterion for judging the quality of the estimator $\widehat{\mathbf{m}}$ in the light of the data is $\Delta h = E_0[d(\widehat{\mathbf{m}}, \widehat{\sigma}^2)]$"[4]. When we apply the expectation to both sides, we obtain the following:

$$\begin{aligned} \Delta h &= E_0\left[d(\widehat{\mathbf{m}}, \widehat{\sigma}^2)\right] \\ &= nE_0\left[\log(2\pi\widehat{\sigma}^2)\right] + n{\sigma_0}^2 E_0\left[\frac{1}{\widehat{\sigma}^2}\right] + E_0\left[\frac{(\mathbf{m} - \widehat{\mathbf{m}})^T(\mathbf{m} - \widehat{\mathbf{m}})}{\widehat{\sigma}^2}\right]. \end{aligned} \tag{2.32}$$

Let $\widehat{\mathbf{m}} = \mathbf{H}\mathbf{y}$ be an unbiased estimator for $\mathbf{m}$ where $\mathbf{H}$ is called the hat matrix. In the context of this thesis, the hat matrix is $\mathbf{H} = \mathbf{X}(\mathbf{X^T X})^{-1}\mathbf{X^T}$, which is a symmetric idempotent matrix. Recall that $E_0(\mathbf{y}) = \mathbf{m}$ by construction; then

$$\mathbf{H}\mathbf{m} = E_0(\widehat{\mathbf{m}}) = \mathbf{m}. \tag{2.33}$$

Since $\mathbf{y} = \mathbf{m} + \boldsymbol{\varepsilon}$ under the true model, we can express the variance estimator as

$$\begin{aligned} \widehat{\sigma}^2 &= \frac{(\mathbf{y} - \widehat{\mathbf{m}})^T(\mathbf{y} - \widehat{\mathbf{m}})}{n} \\ &= \frac{(\mathbf{y} - \mathbf{H}\mathbf{y})^T(\mathbf{y} - \mathbf{H}\mathbf{y})}{n} \\ &= \frac{(\mathbf{m} + \boldsymbol{\varepsilon} - \mathbf{H}(\mathbf{m} + \boldsymbol{\varepsilon}))^T(\mathbf{m} + \boldsymbol{\varepsilon} - \mathbf{H}(\mathbf{m} + \boldsymbol{\varepsilon}))}{n} \\ &= \frac{(\mathbf{m} + \boldsymbol{\varepsilon} - \mathbf{H}\mathbf{m} - \mathbf{H}\boldsymbol{\varepsilon})^T(\mathbf{m} + \boldsymbol{\varepsilon} - \mathbf{H}\mathbf{m} - \mathbf{H}\boldsymbol{\varepsilon})}{n} \\ &= \frac{(\boldsymbol{\varepsilon} - \mathbf{H}\boldsymbol{\varepsilon})^T(\boldsymbol{\varepsilon} - \mathbf{H}\boldsymbol{\varepsilon})}{n} = \frac{\boldsymbol{\varepsilon}^T(\mathbf{I} - \mathbf{H})^T(\mathbf{I} - \mathbf{H})\boldsymbol{\varepsilon}}{n}. \end{aligned} \tag{2.34}$$

At this point Hurvich and Simonoff point out that in practice the unbiasedness assumption for $\widehat{\mathbf{m}}$ will rarely hold. However, they give some justification for doing so which is similar

to the argument used in developing the classical hypothesis tests [4].

In a similar fashion, we can also show that

$$(\mathbf{m} - \widehat{\mathbf{m}})^T(\mathbf{m} - \widehat{\mathbf{m}}) = \boldsymbol{\varepsilon}^T\mathbf{H}^T\mathbf{H}\boldsymbol{\varepsilon}. \tag{2.35}$$

Substituting both (2.34) and (2.35) into (2.32), we obtain the Δh estimator

$$\widetilde{\Delta}h = n\log(2\pi\widehat{\sigma}^2) + n^2E_0\left[\frac{{\sigma_0}^2}{\boldsymbol{\varepsilon}^T\mathbf{B}_1\boldsymbol{\varepsilon}}\right] + nE_0\left[\frac{\boldsymbol{\varepsilon}^T\mathbf{H}^T\mathbf{H}\boldsymbol{\varepsilon}}{\boldsymbol{\varepsilon}^T\mathbf{B}_1\boldsymbol{\varepsilon}}\right], \tag{2.36}$$

where $\mathbf{B}_1 = (\mathbf{I}_n - \mathbf{H})^T(\mathbf{I}_n - \mathbf{H})$ is an $n \times n$ matrix and $\mathbf{I}_n$ is an $n \times n$ identity matrix. Since $\mathbf{H}$ is symmetric idempotent, it can be shown that $\mathbf{B}_1$ is also symmetric idempotent. We then rewrite it using Eigendecomposition, $\mathbf{B}_1 = \boldsymbol{\Gamma}\mathbf{D}\boldsymbol{\Gamma}^{-1}$ where $\mathbf{D}$ is a diagonal matrix of Eigenvalues and $\boldsymbol{\Gamma}$ is the contains the corresponding Eigenvectors in its columns. Since $\mathbf{B}_1$ is symmetric then $\boldsymbol{\Gamma}$ is orthogonal and consequently, $\boldsymbol{\Gamma}^T\boldsymbol{\Gamma} = \mathbf{I}_n$. Using these results, it can be shown that $\mathbf{z} = \boldsymbol{\Gamma}^T\boldsymbol{\varepsilon}/\boldsymbol{\sigma_0}$ is a vector of independent standard normal random variables and

$$\frac{\boldsymbol{\varepsilon}^T B_1\boldsymbol{\varepsilon}}{{\boldsymbol{\sigma_0}}^2} = \mathbf{z}^T\mathbf{D}\mathbf{z} \tag{2.37}$$

is a quadratic form. Let $A_1 = n^2E_0\left[\dfrac{1}{\mathbf{z}^T\mathbf{D}\mathbf{z}}\right]$ so that it takes the place of the second term in (2.36). Then from Jones' 1986 result [5],

$$A_1 = n^2\int_0^1 (1-t)^{r/2-2}\prod_{j=1}^{r}(1 - t + 2td_j)^{-1/2}dt, \tag{2.38}$$

where r is the rank of $\mathbf{B}_1$ and d_j is the j^{th} diagonal element of $\mathbf{D}$[4]. Furthermore, let

$\mathbf{B}_2 = \mathbf{H}^T\mathbf{H}$ and $\mathbf{C} = \mathbf{\Gamma}\mathbf{B}_2\mathbf{\Gamma}^{-1}$. Then,

$$\begin{aligned}\frac{\boldsymbol{\varepsilon}^T\mathbf{H}^T\mathbf{H}\boldsymbol{\varepsilon}}{\boldsymbol{\varepsilon}^T(\mathbf{I}-\mathbf{H})^T(\mathbf{I}-\mathbf{H})\boldsymbol{\varepsilon}} &= \frac{\left(\boldsymbol{\varepsilon}^T\mathbf{B}_2\boldsymbol{\varepsilon}\right)/\sigma_0^2}{\left(\boldsymbol{\varepsilon}^T\mathbf{B}_1\boldsymbol{\varepsilon}\right)/\sigma_0^2} \\ &= \frac{\left(\boldsymbol{\varepsilon}^T\mathbf{\Gamma}\mathbf{\Gamma}^T\mathbf{B}_2\mathbf{\Gamma}^{-1}\mathbf{\Gamma}\boldsymbol{\varepsilon}\right)/\sigma_0^2}{\mathbf{z}^T\mathbf{D}\mathbf{z}} \\ &= \frac{\mathbf{z}^T\mathbf{C}\mathbf{z}}{\mathbf{z}^T\mathbf{D}\mathbf{z}}. \end{aligned} \tag{2.39}$$

using the result $\mathbf{\Gamma}\mathbf{\Gamma}^T = \mathbf{\Gamma}^{-1}\mathbf{\Gamma} = \mathbf{I}_n$. Now let the third term in (2.32) be defined $A_2 = nE_0\left[\dfrac{\mathbf{z}^T\mathbf{C}\mathbf{z}}{\mathbf{z}^T\mathbf{D}\mathbf{z}}\right]$. From Jones' 1987 result [6],

$$A_2 = n\int_0^\infty \sum_{i=1}^{n} \frac{c_{ii}}{1+2d_it}\prod_{i=1}^{n}(1+2d_it)^{-1/2}dt, \tag{2.40}$$

where c_{ii} is the i^{th} diagonal element of $\mathbf{C}$ [4]. Finally, by using (2.38) and (2.40) and dropping the constant term $n\log(2\pi)$, we present the AIC_{C0} criterion:

$$AIC_{C0} = n\log(\boldsymbol{\sigma_0}^2) + A_1 + A_2. \tag{2.41}$$

AIC_{C0} can be found using one-dimensional numerical integration and eigensystem routines. These are sometimes undesirable, especially in our case where SAS uses an exhaustive approach (i.e. it will look at a predetermined number of smoothing parameters defined by the researcher) to find the optimal smoothing parameter [11]. Much like SAS, we will turn to AIC_{C1}, which is void of integrals and matrix decomposition. First, let the distributions of $\mathbf{z}^T\mathbf{C}\mathbf{z}$ and $\mathbf{z}^T\mathbf{D}\mathbf{z}$ be approximated using methods described by Cleveland and Delvin [2]:

$$\mathbf{z}^T\mathbf{D}\mathbf{z} \backsim (\delta_2/\delta_1)\chi^2_{\delta_1^2/\delta_2}$$

and

$$\frac{\mathbf{z}^T\mathbf{C}\mathbf{z}}{\mathbf{z}^T\mathbf{D}\mathbf{z}} \backsim (\nu_1/\delta_1)F_{\nu_1^2/\nu_2,\delta_1^2/\delta_2},$$

where $\delta_1 = tr(\mathbf{B}_1)$, $\delta_2 = tr(\mathbf{B}_1^2)$, $\nu_1 = tr(\mathbf{B}_2)$ and $\nu_2 = tr(\mathbf{B}_2^2)$. The trace of a square matrix is defined as $tr(A) = \sum_{i=1}^{n} a_{ii}$, where a_{ii} are the diagonal elements in the matrix. Furthermore, we use the distributional result that if $X \sim \chi^2$ with ν degrees of freedom, then $\frac{1}{X}$ follows the inverse chi-square distribution with $E\left[\frac{1}{X}\right] = \frac{1}{\nu - 2}$. Consequently, we can approximate the following expectation as

$$E_0\left[\frac{1}{\mathbf{z}^T\mathbf{D}\mathbf{z}}\right] = \left(\frac{\delta_1}{\delta_2}\right)\left(\frac{1}{\delta_1^2/\delta_2 - 2}\right) \tag{2.42}$$

and

$$E_0\left[\frac{\mathbf{z}^T\mathbf{C}\mathbf{z}}{\mathbf{z}^T\mathbf{D}\mathbf{z}}\right] = \frac{\nu_1(\delta_1/\delta_2)}{\delta_1^2/\delta_2 - 2}.$$

Then the proposed criterion for AIC_{C1} as an approximation to AIC_{C0} is

$$AIC_{C1} = n\log(\boldsymbol{\sigma_0}^2) + n\left[\frac{(\nu_1 + n)(\delta_1/\delta_2)}{\delta_1^2/\delta_2 - 2}\right]. \tag{2.43}$$

AIC_{C1} is the final piece in building LOESS. By minimizing AIC_{C1} we are able to find the optimal smoothing parameter using an exhaustive search. In the next chapter we discuss more precisely how this is executed and automated using SAS.

RESULTS

In this chapter we discuss the statistical workflow from data extraction to the final model comparison. Details on how the data was cleaned, processed, and eventually validated using the models are presented.

3.1 Data Formatting

In this study 20 households were sampled from a pool of roughly 800 in the Glasgow EPB smart meter program. The cost of cleaning each household limited this study to 20. Households were chosen randomly, but some were not used based on a count of their records. The baseline for missing readings was set at 20% of the overall days, and in general no more than two consecutive weeks of data were missing from any one household. Most households used in this study contained upwards of 600 records, one for each hour of each day, over a two year period from August 2007 to August 2009. Figure 3.1 shows a screenshot of a sample of raw data.

	A	B	C	D	E	F	G	H
1	Location_ID	Reading	Timestamp_CST	Year	Month	Day	Hour	Minute
4638	9000508	93324.103	8/1/2007 0:00	2007	8	1	0	0
4639	9000508	93327.731	8/1/2007 1:00	2007	8	1	1	0
4640	9000508	93330.915	8/1/2007 2:00	2007	8	1	2	0
4641	9000508	93333.845	8/1/2007 3:00	2007	8	1	3	0
4642	9000508	93336.816	8/1/2007 4:00	2007	8	1	4	0
4643	9000508	93339.557	8/1/2007 5:00	2007	8	1	5	0
4644	9000508	93342.229	8/1/2007 6:00	2007	8	1	6	0
4645	9000508	93345.762	8/1/2007 7:00	2007	8	1	7	0
4646	9000508	93349.707	8/1/2007 8:00	2007	8	1	8	0
4647	9000508	93353.632	8/1/2007 9:00	2007	8	1	9	0
4648	9000508	93357.724	8/1/2007 10:00	2007	8	1	10	0
4649	9000508	93361.839	8/1/2007 11:00	2007	8	1	11	0
4650	9000508	93365.955	8/1/2007 12:00	2007	8	1	12	0

Figure 3.1: Raw data example

Each household is identified by a unique number known as its Location ID. Since we are examining the consumption per hour per household, a reading difference was computed by subtracting the reading of current hour from the next hour. Quality issues

arose due to double readings, skipped readings, and inaccurate readings from the meter. These records were expunged before being considered for modeling. It was clear that a reading was inaccurate based on the magnitude of the reading compared to the others near it and if the sign of the difference between two consecutive readings was not positive.

Of the original explanatory variables proposed in Chapter 2 (temperature, Saturdays, Sundays/holidays) temperature served as the only relevant, easily accessible indicator of consumption in the household. The other variables did not consistently deliver p-values less than 0.05 in preliminary testing, where the autoregressive model was applied to a sample of hours from varying households. Temperature governs how often customers use their heat pumps and air conditioners. Since extreme temperatures prompt individuals to stay indoors, it also explains why consumption from other appliances tends to rise during such times. Temperature is a common thread in most power demand models for these reasons.

	A	B	C	D	E	F	G	H	I	J
1	Location_ID	Reading	Timestamp_CST	Year	Month	Day	Hour	Minute	Reading_Difference	TEMP
2	9000508	93324.1	01AUG07:00:00:00	2007	8	1	0	0	3.63	76.06
3	9000508	93548.68	03AUG07:00:00:00	2007	8	3	0	0	3.41	77.30
4	9000508	93655.35	04AUG07:00:00:00	2007	8	4	0	0	5.02	80.84
5	9000508	93774.49	05AUG07:00:00:00	2007	8	5	0	0	3.97	79.62
6	9000508	93906.75	06AUG07:00:00:00	2007	8	6	0	0	5	81.37
7	9000508	94031	07AUG07:00:00:00	2007	8	7	0	0	4.09	81.76
8	9000508	94146.76	08AUG07:00:00:00	2007	8	8	0	0	4.68	83.64
9	9000508	94276.18	09AUG07:00:00:00	2007	8	9	0	0	4.34	82.20
10	9000508	94398.15	10AUG07:00:00:00	2007	8	10	0	0	4.4	77.96
11	9000508	94511.86	11AUG07:00:00:00	2007	8	11	0	0	3.83	74.20
12	9000508	94625.33	12AUG07:00:00:00	2007	8	12	0	0	4.11	75.60
13	9000508	94733.97	13AUG07:00:00:00	2007	8	13	0	0	4.09	76.30

Figure 3.2: A sample of forecasting data

It was important to gather as much temperature data as possible from the region. The Glasgow EPB had collected some data from a local weather station. We also pursued data from the National Weather Service [10] and sought to merge the two for a more complete set. Glasgow's temperature data was the primary set in the merging process in the event that both sets contained a reading for a single day. Data was available in some

cases by the minute, so this was averaged to get an hourly reading. Once the final data set was prepared, it was merged with the consumption data set by year, month, day, and hour. Then all dates missing either a temperature reading or a power reading were deleted. A screenshot of sample data used for forecasting can be seen in Figure 3.2.

3.2 Autoregressive Forecasting

In Chapter 2, we discussed the autoregressive quadratic model found in literature. In testing, temperature and squared temperature performed well. The model,

$$y_t = \beta_0 + \beta_1 {x_{1t}}^2 + \beta_2 x_{1t} + R_t \tag{3.1}$$

where

$$R_t = \phi_1(R_{t-1}) + ... + \phi_7(R_{t-7}) + \varepsilon_t,$$

and $x_{1t} = hourlytemperature$, differs from the model presented in Chapter 2. In this model, we use an autoregressive error term with up to seven lags. Not every household will have seven, as we will be using SAS's "backstep" option to eliminate lags which are not highly correlated, in which case they will be dropped. The model is simple and will be used as a performance measure by comparing the results to LOESS. Our assumption is that in order for this model to be effective we would have to see the data take on a symmetric U-shape similar to that found in the Florida Power Corporation's data. This is the primary reason for using squared temperature as a predictor. As we can see from Figure 3.3, this will not always be the case. While the U-shape was common amongst the afternoon hours, it was not as prevalent in the early morning where average temperatures rarely climbed above $80°$F. Also, consider that our data comes from an electricity provider. What if a household in the program is using gas for space heating during the winter? This suggests that the symmetric U-shape will not be present coming from such a

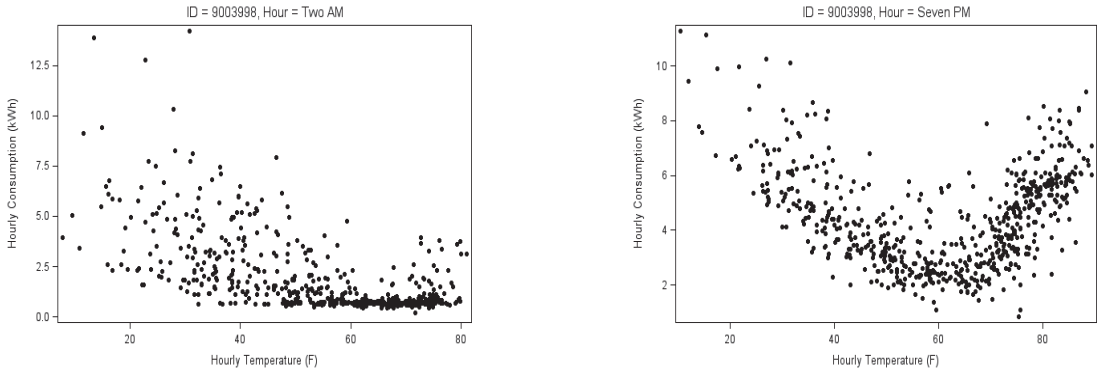

Figure 3.3: Contrasting profiles at different hours of the day.

household, as the demand will be rather flat during this period. Moreover, even when conditions are ideal, the autoregressive model may not outperform LOESS.

3.3 LOESS Forecasting

After we examined the autoregressive model presented by Mendenhall and saw the numerous shapes that demand graphs could take, it seemed natural to have a method that was independent from the shape of the data. LOESS is a one-size-fits-all approach in that we can approximate any demand curve with a series of linear models. This was discussed in great detail in Chapter 2; however, we did not present the actual model used for forecasting. It contains only one variable, temperature. We expect this to fit accurately at the local level where the complexity of the entire data set will not be present. Let,

$$y_t = \beta_0 + \beta_1 x_{1t} + \varepsilon_t, \tag{3.2}$$

where ε_t is an unmodified stochastic error term which follows the assumptions of the OLS method presented in Chapter 2, be the model used for forecasting. We predict this model will be successful in the event that the smoothing parameter $f < 1$. Setting $f = 1$ will produce a weighted simple regression model which may be inferior to the quadratic autoregressive model. Fortunately, this was generally not the case.

3.4 Process Flow

The journey our data takes from extraction to prediction was split into several processes for quality control and debugging purposes. The data is extracted, cleaned, and processed through a series of programs designed to determine the regression parameter estimates. Figure 3.4 sheds light on the differences between the two models. The purpose of each program is discussed. For the autoregressive model, the SAS procedure AUTOREG (cite SAS) is used to calculate the regression-autoregression estimates. We use the "backstep" option to select the lags with p-values below 0.05. We chose seven as the maximum number of lags based on the study from literature discussed at the beginning

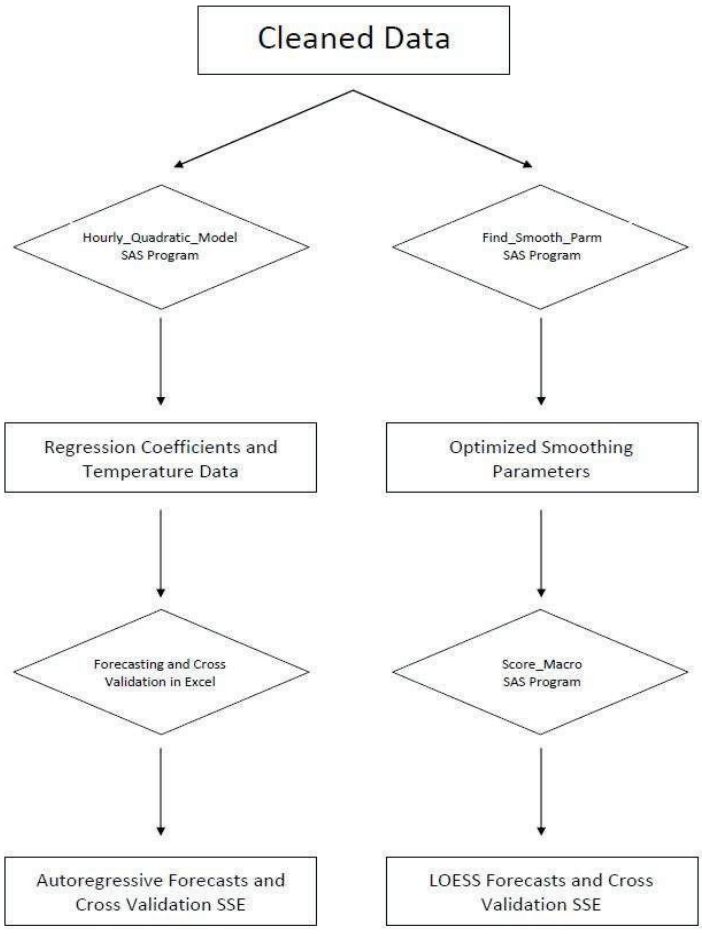

Figure 3.4: A process flow diagram for regression parameter estimation and cross-validation.

of Chapter 2. This leads to the data set in the third tier of Figure 3.4, which contains the temperature values and regression coefficients for every forecasted day.

We knew the Glasgow EPB was interested in forecasting for a household in the program after a specified number of readings were collected; preferably, this time period needs to be as short as possible so they can begin forecasting for the customer. It is unclear how many readings are necessary to produce acceptable predictions, but we chose to begin the process after 90 days of readings had been collected, roughly one seasons worth. After the coefficients had been computed for 90 days worth of data, the 91^{st} point was added, and a new set of regression estimates were computed. This process continued until there were 600 data points; this is the maximum number of data points for most households.

For LOESS, the first step is quite different. The SAS macro "Find Smooth Parm" was written to find the smoothing parameter with the minimum AIC_{C1} by using the "step" option in the LOESS procedure (cite sas). The step option sets the smoothing parameter size to grow by a specified amount from the starting value. The smallest number of data points to start the search was chosen to be seven to guarantee enough unique temperature values needed to have a full-column rank $\mathbf{X}$. Recall that the OLS regression coefficient estimator is

$$\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y}).$$

The inverse of the $\mathbf{X}^T\mathbf{X}$ matrix can be influenced by roundoff error if the temperature readings are close enough together. When weights are added to the equation, such as they are in LOESS, the danger of weighting too many points as zero can also cause singularity. SAS would routinely return an error stating that the matrix was singular if four or fewer data points were used as the starting neighborhood size. The frequency of this error lessened if five points were used and was rarely seen at six. Seven is a conservative starting size selected to keep this error from occurring.

To find the optimal smoothing parameter the AIC_{C1} is computed for every possible smoothing parameter (based on step-size) for every hour of every household. As an

example, the program computes the AIC_{C1} for 90 observations of the midnight hour for a single household. They are ordered from least to greatest, and the record at the top of the list is then selected as the smoothing parameter to use. Then the 91^{st} observation is pulled in, and the process begins again. This method is an exhaustive search, and concerns arose on the runtime of Find Smooth Parm. If the step-size was $\frac{1}{n}$, then the best smoothing parameter could be found. However, there is a tradeoff between the step-size and the speed of the algorithm. We saw it sufficient to grow the neighborhood by seven data points at each step. The final data set seen on the third tier of Figure 3.4 contains the smoothing parameters which are utilized in the next program.

Continuing down the LOESS side of Figure 3.4, the "Score Macro" program is designed to make forecasts based on the smoothing parameters produced from the previous program. Since it is only necessary to forecast one day in advance, we take the model from the previous day's hour and use today's hour's average temperature as input. In reality, this temperature would be a prediction itself from the local weather service. For convenience, we simply use the actual temperature. This concept is described by

$$\hat{y}_i = D_{i-1}(x_i), \tag{3.3}$$

where D_{i-1} is the deterministic part of the previous day's model. The score statement in the LOESS procedure makes this possible. After each day was "scored," as a means of cross-validation the prediction is subtracted from the real reading that hour, and the error and squared error are calculated in SAS. This squared error will be referred to as the cross-validation error. This creates the data set in the fifth and final tier of Figure 3.4.

A similar data set is computed for the autoregressive model, but unfortunately the score statement is not available in the AUTOREG procedure. Each hour's regression coefficients were joined with temperature and reading data. This information was output to Microsoft Excel, and forecasts were made for all 480 hours using the method described

by (3.3). Then error and cross-validation error were computed.

Appendix A contains all SAS code used in processing and forecasting with detailed comments as to how the programs are executed.

3.5 Model Comparison

To see which model performed better, the cross-validation error, or CVE, was transformed by computing the average CVE for the hour, and then taking the square root of the result. This is referred to as the root mean cross-validation error, or RMCVE. This is similar to the root mean squared error, which is the sample standard deviation. This will allow us to express the results in terms of the actual readings, measured in kilowatt hours (kWh). A model indicator was attached to each hour to show which model performed

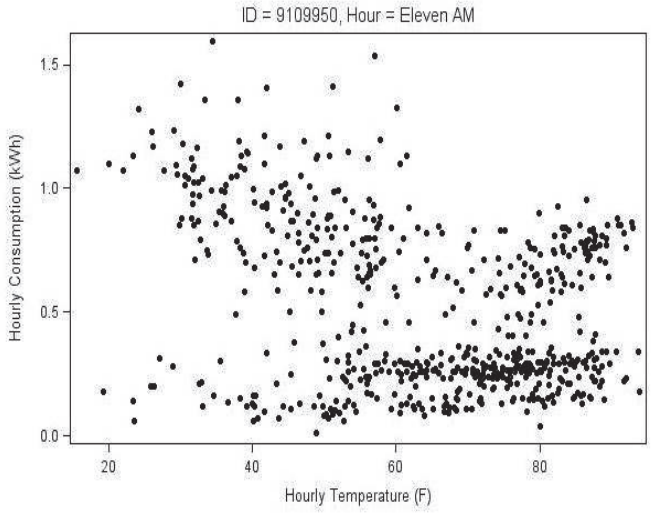

Figure 3.5: The data appears to have two tiers.

better. A "0" indicates LOESS has the lower RMCVE, and a "1" means the autoregressive has the lower RMCVE. As it turns out, LOESS outperforms the autoregressive model in about 80% of all hours. To better understand the performance of each model, the results were aggregated by hour and by household. The full output for each household can be

Table 3.1: Average RMCVE with proportion to LOESS aggregated by hour.

Hour	Autoreg ARMCVE	LOESS ARMCVE	Proportion to LOESS
12:00 AM	0.98	0.92	90%
1:00 AM	0.89	0.83	75%
2:00 AM	0.88	0.80	85%
3:00 AM	0.89	0.81	95%
4:00 AM	0.95	0.86	80%
5:00 AM	1.00	0.92	85%
6:00 AM	1.01	0.99	65%
7:00 AM*	1.09	1.07	70%
8:00 AM	1.12	1.08	75%
9:00 AM	1.21	1.18	70%
10:00 AM	1.26	1.23	75%
11:00 AM	1.27	1.25	70%
12:00 PM	1.26	1.22	70%
1:00 PM	1.25	1.21	80%
2:00 PM	1.24	1.18	80%
3:00 PM	1.26	1.19	90%
4:00 PM	1.28	1.21	80%
5:00 PM	1.24	1.18	85%
6:00 PM	1.27	1.21	90%
7:00 PM	1.28	1.23	85%
8:00 PM	1.28	1.23	75%
9:00 PM	1.22	1.18	75%
10:00 PM	1.09	1.05	75%
11:00 PM	0.97	0.92	75%

*Missing household 9200108 from calculation due to extreme value.

found in Appendix B.

If LOESS is outperforming the autoregressive model in 80% of all hours, then Table 3.1 confirms this assumption if we look at the performance across all hours. It only dips below 70% in one instance. However, if we examine the performance by household (Table 3.2), certain profiles prefer the autoregressive model.

Table 3.2: Average RMCVE with proportion to LOESS aggregated by household.

Location ID	Autoreg ARMCVE	LOESS ARMCVE	Proportion to LOESS
9000508	0.963	0.930	92%
9100456	1.070	1.000	100%
9100720	0.938	0.890	100%
9003998	1.694	1.542	100%
9006578	0.405	0.401	33%
9109645	0.786	0.643	83%
9109922	1.209	1.205	50%
9109950	0.302	0.318	21%
9110002	1.985	1.956	100%
9110003	1.461	1.448	83%
9110004	1.331	1.312	88%
9199920	0.723	0.710	79%
9200108*	1.430	1.398	100%
9200155	0.824	0.793	63%
9200161	0.787	0.804	25%
9200800	1.277	1.086	96%
9201610	1.739	1.611	100%
9202469	0.910	0.891	71%
9202473	1.631	1.597	96%
9202475	1.198	1.124	100%

*Missing 7:00a.m. hour from calculation due to extreme value.

A sign that LOESS might not be performing well is the tendency for the smoothing parameter to converge to $\frac{1}{n}$ as more points are added. This is just a sign that LOESS has a difficult time determining the trend in the data for these households. To further investigate, consider Figure 3.5, which is the typical shape of the data for this location.

For LOESS, the AIC_{C1} in Figure 3.5 is decreasing as more points are being absorbed into the pool for calculation. For the autoregressive model, though, temperature and squared temperature are still relevant explanatory variables. We believed initially the shape of the data was a key factor for the autoregressive model to predict well – this may still be the case, depending on the prediction and how close it is – but it turns out that

LOESS requires this feature even more so. It is reasonable to believe that any number of models, even the mean, would all perform similarly in the situation illustrated in Figure 3.5. For this reason, in practice, we recommend LOESS for all hours and all households.

3.6 The 24-Hour Demand Profile

One of the main objectives of this study is to create a 24-hour demand profile for each household. This can be done by aggregating the predictions for each hour of a single household by date. Figure 3.6 is an example of such a profile.

The line graph is the actual readings, while the blue bars are the predictions given by LOESS. We can see two hours where our predictions were far from the actual value. This is more prevalent for hours where the data is amorphous, such as that of Figure 3.5. The program, "Profile", to generate profiles like this one can be found in Appendix A.

Ideally for the Glasgow EPB, the demand profile should predict a constant demand for electricity. While this will not be possible, it can help consumers realize when their usage is expected to be high, and hopefully there will be some incentive to keep it low during these periods.

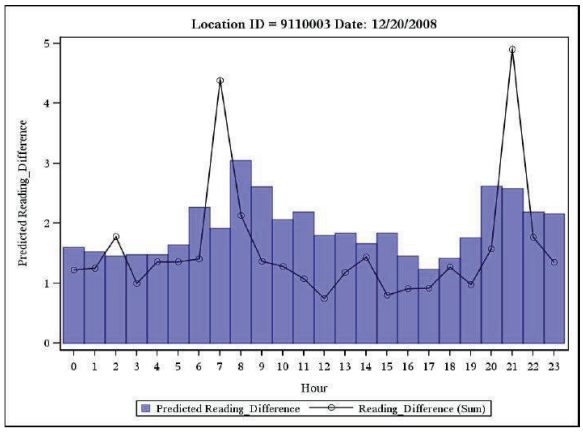

Figure 3.6: A 24 hour demand prediction for Location 9110003 on 12/20/2008.

CONCLUSION

The downfall or success of a simple parametric model, such as the one used for this study, is the amount of variability that can be explained with one predictor. For the autoregressive model, the value of simplicity was overcome by its lack of explanation. It could be possible to design a custom model for each household while still incorporating temperature as the primary predictor. Factors such as insulation grade of the home, thermostat preferences, work schedule, and a host of other tailored variables could be tested for relevance for each program participant. This, of course, was not a luxury we had, and moreover we wanted to see if it could be done differently.

On the other hand, while LOESS finished first, is it a good forecaster of demand? In most cases the RMCVE ranges from 0.5 to 3.0 kilowatt hours (kWh). Given that most people are consuming somewhere between 2 and 10 kWhs per hour, we believe the error is admissible. Ultimately, The Glasgow EPB will have to decide whether this is good enough. At any rate, the outcome of this study shows that forecasting power demand using LOESS should not be overlooked.

It is worth noting that the computational load of finding the smoothing parameter should not be ignored. This harks back to the idea that whatever system is implemented, it should be feasible. If it takes 100 hours to make the forecast for all households in the region each day, then measures must be taken to reduce this time. As an example, for a processor with a clock speed of 2.0 GHz, the minimum smoothing parameter for 90 data points can be found in 0.18 seconds. To find this for 600 data points SAS takes 7.09 seconds. This is partially due to the construction of the algorithm. SAS allows for a user defined starting smoothing parameter and then uses the "step" option to increase the smoothing parameter by a specified amount until it reaches one. This approach lets the neighborhood size vary as more points are added to the data set and keeps the number of steps fixed. We chose to do the opposite and let the number of steps vary, while keeping

the neighborhood size fixed. This creates a greater level of precision when finding the minimum smoothing parameter as more points are added, but the trade-off is run-time. To illustrate further, even if it takes 1 second to make a prediction per household, for 50,000 households it would take 13.89 hours if they were processed sequentially. This is not feasible since customers would not be aware of their demand forecast until roughly 2:00 p.m. Thus, parallel processing is recommended.

Other measures can be implemented to ensure timely forecasts. It might be effective to compute the coefficients for each hour's model and use the model for more than one forecast. It was expressed that Glasgow would not be tracking weekends as part of the program. This time could be spent computing a model to be used for forecasting demand for the entire week using temperature data as it becomes available. This would bypass the time consuming process of finding the optimal smoothing parameter. A future study could be done to determine the maximum amount of days one could use the coefficients before forecasts become too deviant.

One last piece of information to relay is a warning to SAS users. SAS will not allow LOESS to extrapolate for those temperatures which fall outside the range of values already used to form the model. This is not the case for the AUTOREG procedure. The 90-data-point minimum does provide a wide range of temperatures, but only because of the time period in which it was taken. The data is collected starting in late August, a month that produces some of the highest temperatures during the year, and ends with November, which is traditionally much cooler. As new temperatures are absorbed that fall outside the current range, they become the new maximum/minimum temperature. Fortunately, we only see around 10 cases out of roughly 500 per hour where the prediction is missing.

APPENDIX A

```
/*-------------------------------------------------------------------------*
 * Macro Name: Find_Smooth_Parm                                            *
 * Author: Craig Dickson                                                   *
 * Purpose: Find the optiomal smoothing parameter for each hour            *
 * Input: n = Starting hour in the list                                   *
 * index = last hour in the list                                  *
 * base = minimum number of points to be used                     *
 * max = maximum number of points to be used                      *
 * ghourly._24hour_sets = List of Hours for each household        *
 * Output: ghourly.&name_AICC1 which contains:                             *
 * Year, Month, Day, Hour, Temperature, Actual Power Reading,     *
 * Smoothing Parameter to be used, Data points used, AICC1    *
 *-------------------------------------------------------------------------*/


libname ghourly '\\vmware-host\Shared Folders\gstudent documents\desktop
\Craigs Folder\Thesis\Data\SAS Data\Glasgow Hourly';

%macro Find_Smooth_Parm(n,index,base,max);
/*--Outer Loop--*/
%DO %WHILE(&n<=&index);

data _NULL_;
    obsnum=&n;
set ghourly._24hour_sets point=obsnum; /*Points to the name of the current hour in the list*/
call symput('name',compress(filenames));
/*Creates a macro variable name from filenames in _24hour_set.
 An example of this would be h09000508_ten_am*/
output;
stop;
run;

proc sort data=ghourly.&name out=&name; /*Sorts tha data for quality assurance*/
by Timestamp_CST;
run;

%let loopcounter=1; /*Counter used to identify the first iteration*/
%let obscounter=&base; /*keeps count of the current number of points used for LOESS*/
/*--Inner Loop--*/
%DO %WHILE(&obscounter<=&max);

ods output FitSummary=&name._Summary; /*Outputs a fit summary for LOESS*/

%let step=%sysevalf(7/&obscounter); /*Defines the step size*/
%let start=%sysevalf(1*&step);
     /*Defines the starting neighborhood size in terms of the step size*/

proc loess data=&name (obs=&obscounter);
    /*LOESS, computing for the number of points specified by obscounter*/
model Reading_Difference=Temp / smooth=&start to 1 by &step dfmethod=exact r clm;
run;

proc SQL OUTOBS=1; /*Grabs the smoothing parameter with the minimum AICC1*/
CREATE TABLE Min_AICC1 AS
SELECT *
FROM &name._summary
WHERE Label1="AICC1"
ORDER BY cValue1;
quit;

%if &loopcounter=1 %then %do;
    /*Creates the data set which holds the smoothing parameters,at the
    first iteration this will be the smoothing parameter for &base points*/
data &name._AICC1;
set Min_AICC1;
n=&obscounter;
     /*Creates a column which stores the current number of points used for LOESS*/
run;
```

```
%end; %else %do;
data currentobs;
      /*stores the current number of points*/
n=&obscounter;
run;

data Min_AICC1;
       /*Merges the number of points used with the smoothing parameter for that many points*/
merge Min_AICC1 currentobs;
run;

data &name._AICC1; /*Sets the current calculations to those before it*/
set &name._AICC1 Min_AICC1;
run;
%end;

%let obscounter=%eval(&obscounter+1);
    /*Prepares the program to absorb the next data point and compute the new model*/
%let loopcounter=%eval(&loopcounter+1);

%END;
/*--Closes Inner Loop--*/
data ghourly.&name._AICC1;
     /*Stores the temporary data set for the current hour into the SAS library ghourly*/
merge &name._AICC1 ghourly.&name (firstobs=&base obs=&max);
Filenames="&name";
keep Filenames Year Month Day Hour Temp Reading_Difference SmoothingParameter n cValue1;
output;
run;

%let n=%eval(&n+1);
/*Increments n which will grab the next hour in the list. For example,
if the current hour was Ten AM then the next one is Eleven AM*/

%END;
/*--Closes Outer Loop--*/
%mend Find_Smooth_Parm;

%Find_Smooth_Parm(1,480,90,600); /*user input*/
```

```
/*-----------------------------------------------------------------------------*
 * Macro Name: Hourly_Quadratic_Model                                          *
 * Author: Craig Dickson                                                       *
 * Purpose: Compute the regression coefficients for the parametric model       *
 * Input: n = Starting hour in the list                                       *
 * index = last hour in the list                                  *
 * base = minimum number of points to be used                     *
 * max = maximum number of points to be used                      *
 * ghourly._24hour_sets = List of Hours for each household        *
 * Output: ghourly.&name_autoreg which contains:                               *
 * SSE, intercept term, coefficient for temperature squared,      *
 * coefficient for temperaure, data points used, actual reading   *
 * AICC, temperature                                 *
 *-----------------------------------------------------------------------------*/

libname ghourly '\\vmware-host\Shared Folders\gstudent documents\
desktop\Craigs Folder\Thesis\Data\SAS Data\Glasgow Hourly';

%macro qm_autoreg(n,index,base,max);
/*--Outer Loop--*/
%DO %WHILE(&n<=&index);

data _NULL_;
    obsnum=&n;
set ghourly._24hour_sets point=obsnum; /*Points to the name of the current hour in the list*/
call symput('name',compress(filenames));
       /*Creates a macro variable name from filenames in _24hour_set.
 An example of this would be h09000508_ten_am*/
output;
stop;
run;

proc sort data=ghourly.&name out=&name; /*Sorts tha data for quality assurance*/
by Timestamp_CST;
run;

data &name; /*Squares the temperature*/
set &name;
temp_sq=temp*temp;
output;
run;

%let loopcounter=1; /*Counter used to identify the first iteration*/
%let obscounter=&base; /*keeps count of the current number of points used for LOESS*/
/*--Inner Loop--*/
%DO %WHILE(&obscounter<=&max);

ods output fitsummary=fitsum; /*Outputs a fit summary for AUTOREG*/

proc autoreg data=&name (obs=&obscounter) outest=est;
        /*AUTOREG, computing for number of points specified by obscounter*/
model Reading_Difference = temp_sq temp / dw=7 dwprob method=ml nlag=7 backstep;
         /*backstep for lag selection*/
run;

proc SQL OUTOBS=1; /*Finds the AICC for each model*/
CREATE TABLE Min_AICC1 AS
SELECT Label2, cValue2
FROM fitsum
WHERE Label2="AICC"
ORDER BY cValue2;
quit;

%if &loopcounter=1 %then %do;
             /*Creates the set which will contain the AICC's*/
data &name._AICC1;
set Min_AICC1;
n=&obscounter;
```

```
              /*Creates a column which stores the current number of points used for AUTOREG*/
output;
run;

data Q_Model; /*Creates data set which holds regression coefficients*/
set est (rename = (_SSE_=SSE_Autoreg temp_sq=Temp_sq_Coeff temp=Temp_Coeff));
n=&obscounter;
keep n SSE_Autoreg Intercept Temp_sq_Coeff Temp_Coeff;
output;
run;
%end; %else %do;
data currentobs;/*stores the current number of points*/
n=&obscounter;
output;
run;

data Min_AICC1; /*Merges the number of points used by AUTOREG with AICC data*/
merge Min_AICC1 currentobs;
output;
run;

data &name._AICC1; /*Adds more observations to the AICC set*/
set &name._AICC1 Min_AICC1;
output;
run;

data est; /*merges the currentobs set with the est set created by AUTOREG*/
merge est currentobs;
output;
run;

data Q_Model; /*Adds more observations to the Q_Model set*/
set Q_Model est (rename = (_SSE_=SSE_Autoreg temp_sq=Temp_sq_Coeff temp=Temp_Coeff));
keep n SSE_Autoreg Intercept Temp_sq_Coeff Temp_Coeff;
output;
run;
%end;

%let obscounter=%eval(&obscounter+1);
        /*Prepares the program to absorb the next data point and compute the new model*/
%let loopcounter=%eval(&loopcounter+1);

%END;
/*--Closes Inner Loop--*/
data &name;
     /*Grabs the temperature and readings data from the original data set
       between the base and max observations*/
set &name (firstobs=&base obs=&max);
keep reading_difference temp;
output;
run;

data ghourly.&name._autoreg;
        /*Merges Q_Model, AICC, and &name to form the output set in ghourly*/
merge Q_Model &name._AICC1 (rename = (cValue2=AICC_AR)) &name;
drop Label2;
output;
run;

%let n=%eval(&n+1);
  /*Increments n which will grab the next hour in the list. For example,
   if the current hour was Ten AM then the next one is Eleven AM*/

%END;
/*--Closes Outer Loop--*/
%mend qm_autoreg;

%qm_autoreg(1,480,90,600); /*user input*/
```

```
/*-------------------------------------------------------------------------*
 * Macro Name: Score_Macro                                                 *
 * Author: Craig Dickson                                                   *
 * Purpose: Make the power demand forecast for each day                *
 * Input: n = Starting hour in the list                                *
 * index = last hour in the list                                       *
 * base = minimum number of points to be used                          *
 * max = maximum number of points to be used                           *
 * ghourly._24hour_sets = List of Hours for each household             *
 * ghourly.&name_AICC1 = Contains smoothing parameters for LOESS       *
 * Output: ghourly.&name_prediction which contains:                    *
 * Year, Month, Day, Hour, Temperature, Actual Power Reading,      *
 * Smoothing Parameter to be used, Data points used,               *
 * Forecasted Reading, Residual, Residual Squared                  *
 *-------------------------------------------------------------------------*/

libname ghourly '\\vmware-host\Shared Folders\gstudent documents\desktop
\Craigs Folder\Thesis\Data\SAS Data\Glasgow Hourly';

%macro Score_Macro(n,index,base,max);
/*--Outer Loop--*/
%DO %WHILE(&n<=&index);

data _NULL_;
    obsnum=&n;
set ghourly._24hour_sets point=obsnum;
    /*Points to the name of the current hour in the list*/
call symput('name',compress(filenames));
   /*Creates a macro variable name from filenames in _24hour_set.
An example of this would be h09000508_ten_am*/
output;
stop;
run;

%let loopcounter=1; /*Counter used to identify the first iteration*/
%let obscounter=&base; /*keeps count of the current number of points used for LOESS*/
/*--Inner Loop--*/
%DO %WHILE(&obscounter<&max);

data currentsp;
   /*Grabs the smoothing parameter for the current number of points used for LOESS*/
obsnum=&loopcounter;
set ghourly.&name._AICC1 point=obsnum;
call symput('sp',SmoothingParameter);
output;
stop;
run;

data futuretemp;
   /*Grabs the next day's temperature as the temperature to forecast today's demand*/
obsnum=&loopcounter+1;
set ghourly.&name._AICC1 point=obsnum;
output;
stop;
run;

ods output FitSummary=&name._Summary /*Outputs a fit summary for LOESS*/
   ScoreResults=&name._scored; /*Outputs the results of the forecasted demand*/

proc loess data=ghourly.&name (obs=&obscounter);
   /*LOESS, computing the model with the optimal smoothing parameter*/
model Reading_Difference=Temp / smooth=&sp dfmethod=exact r clm;
score data=futuretemp ID=(Temp); /*Score statement which gives the forecasted demand*/
run;

data &name._scored; /*Computes the resiedual and residual squared from the score results*/
set &name._scored;
residual=Reading_Difference-p_Reading_Difference;
```

```
residual_sq=(Reading_Difference-p_Reading_Difference)**2;
drop scoredata obs;
output;
run;

%if &loopcounter=1 %then %do;
    /*Creates a data set which stores the scored data*/
data &name._prediction;
set &name._scored;
run;
%end; %else %do;
data &name._prediction;
    /*Adds new results to the prediction data set*/
set &name._prediction &name._scored;
run;
%end;

%let obscounter=%eval(&obscounter+1);
    /*Prepares the program to absorb the next data point and compute the new model*/
%let loopcounter=%eval(&loopcounter+1);

%END;
/*--Closes Inner Loop--*/
data ghourly.&name._prediction;
   /*Outputs the predictions to the SAS library ghourly*/
set &name._prediction;
output;
run;

%let n=%eval(&n+1);
 /*Increments n which will grab the next hour in the list. For example,
 if the current hour was Ten AM then the next one is Eleven AM*/

%END;
/*--Closes Outer Loop--*/
%mend Score_Macro;

%Score_Macro(1,480,90,600); /*user input*/
```

```
/*-----------------------------------------------------------------------------*
 * Macro Name: Get_Profile                                                     *
 * Author: Craig Dickson                                                       *
 * Purpose: produce a 24 demand profile graph                                  *
 * Input: n = Starting hour in the list                                        *
 * index = last hour in the list                                               *
 * month = selected month                                                      *
 * day = selected day                                                          *
 * year = selected year                                                        *
 * ghourly._24hour_sets = List of Hours for each household                     *
 * ghourly.&name_prediction                                                    *
 * Output: A bar graph of the predicted demand in kilowatt hours               *
 * A Line plot of actual readings which overlays the bar graph                 *
 *-----------------------------------------------------------------------------*/


libname ghourly '\\vmware-host\Shared Folders\gstudent documents\desktop
         \Craigs Folder\Thesis\Data\SAS Data\Glasgow Hourly';

%macro get_profile(n, index, month, day, year);

%let loopcounter=1; /*Counter used to identify the first iteration*/
/*--Begin Loop--*/
%DO %WHILE(&n<=&index);

data _NULL_;
    obsnum=&n;
set ghourly._24hour_sets point=obsnum;
     /*Points to the name of the current hour in the list*/
call symput('name',compress(filenames));
    /*Creates a macro variable name from filenames in _24hour_set.
An example of this would be h09000508_ten_am*/
output;
stop;
run;

data grab_row;
    /*Grabs the row which contains the actual and predicted readings*/
set ghourly.&name._prediction;
where Year=&year and Month=&month and Day=&day;
output;
run;

%if &loopcounter=1 %then %do;
    /*Creates the profile data set with grab row*/
data profile;
set grab_row;
output;
run;
%end; %else %do; /*Adds hours to the profile set*/
data profile;
set profile grab_row;
output;
run;
%end;

%let loopcounter=%eval(&loopcounter+1);
%let n=%eval(&n+1);
/*Increments n which will grab the next hour in the list. For example,
 if the current hour was Ten AM then the next one is Eleven AM*/

%END;
/*--Closes Loop--*/
ODS PDF FILE="\\vmware-host\Shared Folders\gstudent documents\desktop
              \Craigs Folder\Thesis\SAS Stuff\Plots\profileplot.pdf";
proc sgplot data=profile; /*Creates the output plots*/
title "Location ID = 9110003 Date: &month./&day./&year";
VBAR Hour/ RESPONSE=p_Reading_Difference BARWIDTH=1 FILLATTRS=(COLOR=blue) TRANSPARENCY=.5;
```

```
VLINE Hour/ RESPONSE=Reading_Difference LINEATTRS=(COLOR=black) BREAK MARKERS;
run;
quit;
ODS PDF CLOSE;

%mend get_profile;

%get_profile(1,24,12,20,2008); /*user input*/
```

APPENDIX B

Table 6.1: Root mean cross-validation error (kWh) model comparison: 9000508

Location ID	Hour	Autoreg RMCVE	LOESS RMCVE	Model Indicator
9000508	12:00 AM	0.795	0.663	0
	1:00 AM	0.596	0.571	0
	2:00 AM	0.716	0.555	0
	3:00 AM	0.659	0.532	0
	4:00 AM	0.605	0.512	0
	5:00 AM	0.579	0.502	0
	6:00 AM	0.733	0.734	1
	7:00 AM	0.696	0.689	0
	8:00 AM	0.880	0.871	0
	9:00 AM	1.055	1.041	0
	10:00 AM	1.032	1.021	0
	11:00 AM	1.146	1.125	0
	12:00 PM	1.072	1.058	0
	1:00 PM	1.081	1.071	0
	2:00 PM	1.075	1.062	0
	3:00 PM	1.057	1.045	0
	4:00 PM	1.080	1.061	0
	5:00 PM	1.158	1.143	0
	6:00 PM	1.314	1.300	0
	7:00 PM	1.285	1.285	0
	8:00 PM	1.318	1.316	0
	9:00 PM	1.266	1.267	1
	10:00 PM	1.071	1.069	0
	11:00 PM	0.842	0.822	0

Table 6.2: Root mean cross-validation error (kWh) model comparison: 9100456

Location ID	Hour	Autoreg RMCVE	LOESS RMCVE	Model Indicator
9100456	12:00 AM	1.008	0.913	0
	1:00 AM	0.942	0.846	0
	2:00 AM	0.948	0.847	0
	3:00 AM	0.958	0.865	0
	4:00 AM	0.971	0.888	0
	5:00 AM	0.952	0.873	0
	6:00 AM	0.819	0.773	0
	7:00 AM	0.905	0.842	0
	8:00 AM	0.927	0.859	0
	9:00 AM	0.989	0.932	0
	10:00 AM	1.178	1.098	0
	11:00 AM	1.147	1.096	0
	12:00 PM	1.224	1.177	0
	1:00 PM	1.217	1.185	0
	2:00 PM	1.186	1.132	0
	3:00 PM	1.243	1.194	0
	4:00 PM	1.344	1.299	0
	5:00 PM	1.288	1.218	0
	6:00 PM	1.167	1.090	0
	7:00 PM	1.097	1.041	0
	8:00 PM	1.059	0.991	0
	9:00 PM	1.078	1.006	0
	10:00 PM	1.061	0.957	0
	11:00 PM	0.979	0.888	0

Table 6.3: Root mean cross-validation error (kWh) model comparison: 9100720

Location ID	Hour	Autoreg RMCVE	LOESS RMCVE	Model Indicator
9100720	12:00 AM	0.850	0.744	0
	1:00 AM	0.656	0.582	0
	2:00 AM	0.597	0.520	0
	3:00 AM	0.576	0.536	0
	4:00 AM	0.446	0.431	0
	5:00 AM	0.458	0.443	0
	6:00 AM	0.628	0.591	0
	7:00 AM	0.815	0.760	0
	8:00 AM	0.829	0.780	0
	9:00 AM	0.898	0.861	0
	10:00 AM	1.032	0.994	0
	11:00 AM	1.194	1.155	0
	12:00 PM	1.233	1.183	0
	1:00 PM	1.190	1.143	0
	2:00 PM	1.177	1.135	0
	3:00 PM	1.231	1.188	0
	4:00 PM	1.106	1.059	0
	5:00 PM	1.116	1.064	0
	6:00 PM	1.140	1.092	0
	7:00 PM	1.211	1.183	0
	8:00 PM	1.133	1.083	0
	9:00 PM	1.198	1.158	0
	10:00 PM	1.003	0.934	0
	11:00 PM	0.797	0.732	0

Table 6.4: Root mean cross-validation error (kWh) model comparison: 9003998

Location ID	Hour	Autoreg RMCVE	LOESS RMCVE	Model Indicator
9003998	12:00 AM	1.473	1.249	0
	1:00 AM	1.495	1.260	0
	2:00 AM	1.853	1.506	0
	3:00 AM	2.501	2.152	0
	4:00 AM	3.923	3.110	0
	5:00 AM	4.232	3.744	0
	6:00 AM	1.695	1.681	0
	7:00 AM	1.801	1.778	0
	8:00 AM	1.464	1.409	0
	9:00 AM	1.343	1.306	0
	10:00 AM	1.424	1.382	0
	11:00 AM	1.526	1.443	0
	12:00 PM	1.556	1.461	0
	1:00 PM	1.498	1.435	0
	2:00 PM	1.436	1.381	0
	3:00 PM	1.329	1.263	0
	4:00 PM	1.435	1.347	0
	5:00 PM	1.356	1.289	0
	6:00 PM	1.196	1.126	0
	7:00 PM	1.219	1.195	0
	8:00 PM	1.463	1.400	0
	9:00 PM	1.301	1.222	0
	10:00 PM	1.071	0.984	0
	11:00 PM	1.062	0.883	0

Table 6.5: Root mean cross-validation error (kWh) model comparison: 9006578

Location ID	Hour	Autoreg RMCVE	LOESS RMCVE	Model Indicator
9006578	12:00 AM	0.121	0.137	1
	1:00 AM	0.125	0.129	1
	2:00 AM	0.119	0.121	1
	3:00 AM	0.111	0.005	0
	4:00 AM	0.106	0.121	1
	5:00 AM	0.121	0.141	1
	6:00 AM	0.167	0.179	1
	7:00 AM	0.288	0.310	1
	8:00 AM	0.225	0.243	1
	9:00 AM	0.396	0.445	1
	10:00 AM	0.576	0.586	1
	11:00 AM	0.595	0.592	0
	12:00 PM	0.515	0.520	1
	1:00 PM	0.544	0.568	1
	2:00 PM	0.497	0.522	1
	3:00 PM	0.579	0.563	0
	4:00 PM	0.707	0.624	0
	5:00 PM	0.691	0.653	0
	6:00 PM	0.766	0.726	0
	7:00 PM	0.743	0.728	0
	8:00 PM	0.684	0.653	0
	9:00 PM	0.559	0.571	1
	10:00 PM	0.317	0.326	1
	11:00 PM	0.160	0.169	1

Table 6.6: Root mean cross-validation error (kWh) model comparison: 9109645

Location ID	Hour	Autoreg RMCVE	LOESS RMCVE	Model Indicator
9109645	12:00 AM	0.569	0.494	0
	1:00 AM	0.482	0.442	0
	2:00 AM	0.425	0.426	1
	3:00 AM	0.405	0.398	0
	4:00 AM	0.373	0.382	1
	5:00 AM	0.358	0.369	1
	6:00 AM	0.421	0.430	1
	7:00 AM	0.482	0.452	0
	8:00 AM	0.643	0.553	0
	9:00 AM	0.786	0.688	0
	10:00 AM	0.975	0.870	0
	11:00 AM	1.081	0.973	0
	12:00 PM	0.948	0.809	0
	1:00 PM	0.972	0.770	0
	2:00 PM	1.103	0.821	0
	3:00 PM	1.082	0.778	0
	4:00 PM	1.147	0.821	0
	5:00 PM	1.280	1.033	0
	6:00 PM	1.179	0.881	0
	7:00 PM	1.052	0.736	0
	8:00 PM	0.892	0.645	0
	9:00 PM	0.845	0.656	0
	10:00 PM	0.707	0.526	0
	11:00 PM	0.659	0.471	0

Table 6.7: Root mean cross-validation error (kWh) model comparison: 9109922

Location ID	Hour	Autoreg RMCVE	LOESS RMCVE	Model Indicator
9109922	12:00 AM	0.753	0.751	0
	1:00 AM	0.577	0.592	1
	2:00 AM	0.569	0.559	0
	3:00 AM	0.634	0.624	0
	4:00 AM	0.613	0.608	0
	5:00 AM	1.069	0.991	0
	6:00 AM	1.076	1.041	0
	7:00 AM	0.990	0.996	1
	8:00 AM	1.133	1.131	0
	9:00 AM	1.312	1.316	1
	10:00 AM	1.362	1.353	0
	11:00 AM	1.192	1.203	1
	12:00 PM	1.210	1.211	1
	1:00 PM	1.138	1.150	1
	2:00 PM	1.207	1.215	1
	3:00 PM	1.357	1.339	0
	4:00 PM	1.499	1.508	1
	5:00 PM	1.712	1.652	0
	6:00 PM	1.694	1.687	0
	7:00 PM	1.953	1.961	1
	8:00 PM	1.961	1.988	1
	9:00 PM	1.738	1.777	1
	10:00 PM	1.267	1.277	1
	11:00 PM	0.995	0.981	0

Table 6.8: Root mean cross-validation error (kWh) model comparison: 9109950

Location ID	Hour	Autoreg RMCVE	LOESS RMCVE	Model Indicator
9109950	12:00 AM	0.329	0.327	0
	1:00 AM	0.325	0.334	1
	2:00 AM	0.315	0.301	0
	3:00 AM	0.307	0.302	0
	4:00 AM	0.299	0.296	0
	5:00 AM	0.323	0.306	0
	6:00 AM	0.251	0.264	1
	7:00 AM	0.255	0.262	1
	8:00 AM	0.258	0.270	1
	9:00 AM	0.266	0.276	1
	10:00 AM	0.308	0.334	1
	11:00 AM	0.323	0.355	1
	12:00 PM	0.312	0.342	1
	1:00 PM	0.328	0.350	1
	2:00 PM	0.332	0.357	1
	3:00 PM	0.324	0.343	1
	4:00 PM	0.320	0.336	1
	5:00 PM	0.316	0.342	1
	6:00 PM	0.319	0.346	1
	7:00 PM	0.314	0.349	1
	8:00 PM	0.190	0.210	1
	9:00 PM	0.309	0.332	1
	10:00 PM	0.325	0.366	1
	11:00 PM	0.307	0.321	1

Table 6.9: Root mean cross-validation error (kWh) model comparison: 9110002

Location ID	Hour	Autoreg RMCVE	LOESS RMCVE	Model Indicator
9110002	12:00 AM	0.923	0.912	0
	1:00 AM	0.912	0.906	0
	2:00 AM	0.823	0.807	0
	3:00 AM	0.804	0.784	0
	4:00 AM	0.835	0.828	0
	5:00 AM	1.115	1.113	0
	6:00 AM	3.803	3.761	0
	7:00 AM	3.112	3.050	0
	8:00 AM	2.361	2.303	0
	9:00 AM	2.983	2.945	0
	10:00 AM	3.200	3.153	0
	11:00 AM	3.040	2.994	0
	12:00 PM	2.715	2.690	0
	1:00 PM	2.253	2.220	0
	2:00 PM	1.993	1.992	0
	3:00 PM	1.940	1.927	0
	4:00 PM	2.177	2.151	0
	5:00 PM	2.182	2.168	0
	6:00 PM	2.109	2.086	0
	7:00 PM	2.078	2.060	0
	8:00 PM	1.911	1.891	0
	9:00 PM	1.730	1.648	0
	10:00 PM	1.377	1.314	0
	11:00 PM	1.260	1.241	0

Table 6.10: Root mean cross-validation error (kWh) model comparison: 9110003

Location ID	Hour	Autoreg RMCVE	LOESS RMCVE	Model Indicator
9110003	12:00 AM	1.284	1.274	0
	1:00 AM	0.961	0.942	0
	2:00 AM	0.773	0.767	0
	3:00 AM	0.740	0.731	0
	4:00 AM	0.704	0.656	0
	5:00 AM	0.712	0.686	0
	6:00 AM	1.194	1.185	0
	7:00 AM	1.386	1.381	0
	8:00 AM	1.529	1.505	0
	9:00 AM	1.569	1.571	1
	10:00 AM	1.420	1.418	0
	11:00 AM	1.586	1.556	0
	12:00 PM	1.670	1.664	0
	1:00 PM	1.737	1.717	0
	2:00 PM	1.812	1.792	0
	3:00 PM	1.891	1.889	0
	4:00 PM	1.784	1.789	1
	5:00 PM	1.604	1.594	0
	6:00 PM	1.759	1.742	0
	7:00 PM	1.870	1.843	0
	8:00 PM	2.139	2.140	1
	9:00 PM	1.923	1.919	0
	10:00 PM	1.596	1.596	1
	11:00 PM	1.418	1.401	0

Table 6.11: Root mean cross-validation error (kWh) model comparison: 9110004

Location ID	Hour	Autoreg RMCVE	LOESS RMCVE	Model Indicator
9110004	12:00 AM	1.017	0.990	0
	1:00 AM	0.756	0.725	0
	2:00 AM	0.690	0.659	0
	3:00 AM	0.631	0.601	0
	4:00 AM	0.753	0.733	0
	5:00 AM	1.075	1.048	0
	6:00 AM	1.240	1.225	0
	7:00 AM	1.391	1.386	0
	8:00 AM	1.472	1.477	1
	9:00 AM	1.530	1.514	0
	10:00 AM	1.491	1.473	0
	11:00 AM	1.501	1.500	0
	12:00 PM	1.359	1.365	1
	1:00 PM	1.265	1.241	0
	2:00 PM	1.208	1.185	0
	3:00 PM	1.242	1.220	0
	4:00 PM	1.213	1.208	0
	5:00 PM	1.406	1.396	0
	6:00 PM	1.822	1.768	0
	7:00 PM	2.074	2.060	0
	8:00 PM	2.067	2.070	1
	9:00 PM	1.773	1.745	0
	10:00 PM	1.682	1.636	0
	11:00 PM	1.280	1.256	0

Table 6.12: Root mean cross-validation error (kWh) model comparison: 9199920

Location ID	Hour	Autoreg RMCVE	LOESS RMCVE	Model Indicator
9199920	12:00 AM	0.855	0.837	0
	1:00 AM	0.693	0.698	1
	2:00 AM	0.525	0.511	0
	3:00 AM	0.482	0.474	0
	4:00 AM	0.415	0.412	0
	5:00 AM	0.461	0.461	0
	6:00 AM	0.487	0.501	1
	7:00 AM	0.467	0.483	1
	8:00 AM	0.510	0.531	1
	9:00 AM	0.591	0.593	1
	10:00 AM	0.779	0.770	0
	11:00 AM	0.992	0.980	0
	12:00 PM	0.949	0.925	0
	1:00 PM	0.939	0.933	0
	2:00 PM	0.792	0.784	0
	3:00 PM	0.803	0.789	0
	4:00 PM	0.894	0.861	0
	5:00 PM	0.907	0.854	0
	6:00 PM	1.048	1.012	0
	7:00 PM	0.798	0.740	0
	8:00 PM	0.781	0.748	0
	9:00 PM	0.717	0.689	0
	10:00 PM	0.661	0.643	0
	11:00 PM	0.816	0.805	0

Table 6.13: Root mean cross-validation error (kWh) model comparison: 9200108

Location ID	Hour	Autoreg RMCVE	LOESS RMCVE	Model Indicator
9200108	12:00 AM	1.489	1.469	0
	1:00 AM	1.618	1.577	0
	2:00 AM	1.583	1.543	0
	3:00 AM	1.608	1.556	0
	4:00 AM	1.489	1.435	0
	5:00 AM	1.295	1.250	0
	6:00 AM	1.221	1.185	0
	7:00 AM	721.325	717.775	0
	8:00 AM	1.408	1.359	0
	9:00 AM	1.587	1.550	0
	10:00 AM	1.342	1.299	0
	11:00 AM	1.214	1.198	0
	12:00 PM	1.178	1.175	0
	1:00 PM	1.286	1.286	0
	2:00 PM	1.343	1.343	0
	3:00 PM	1.424	1.407	0
	4:00 PM	1.430	1.394	0
	5:00 PM	1.422	1.391	0
	6:00 PM	1.570	1.554	0
	7:00 PM	1.454	1.420	0
	8:00 PM	1.592	1.582	0
	9:00 PM	1.501	1.475	0
	10:00 PM	1.437	1.407	0
	11:00 PM	1.335	1.310	0

Table 6.14: Root mean cross-validation error (kWh) model comparison: 9200155

Location ID	Hour	Autoreg RMCVE	LOESS RMCVE	Model Indicator
9200155	12:00 AM	0.932	0.911	0
	1:00 AM	0.969	0.729	0
	2:00 AM	0.976	0.764	0
	3:00 AM	0.993	0.837	0
	4:00 AM	1.090	0.988	0
	5:00 AM	1.046	0.937	0
	6:00 AM	0.660	0.649	0
	7:00 AM	0.796	0.800	1
	8:00 AM	0.998	1.002	1
	9:00 AM	0.917	0.911	0
	10:00 AM	0.709	0.717	1
	11:00 AM	0.603	0.613	1
	12:00 PM	0.731	0.714	0
	1:00 PM	0.747	0.744	0
	2:00 PM	0.615	0.630	1
	3:00 PM	0.625	0.640	1
	4:00 PM	0.622	0.629	1
	5:00 PM	0.630	0.667	1
	6:00 PM	0.720	0.718	0
	7:00 PM	0.998	0.994	0
	8:00 PM	1.017	1.015	0
	9:00 PM	1.062	1.051	0
	10:00 PM	0.871	0.869	0
	11:00 PM	0.459	0.508	1

Table 6.15: Root mean cross-validation error (kWh) model comparison: 9200161

Location ID	Hour	Autoreg RMCVE	LOESS RMCVE	Model Indicator
9200161	12:00 AM	0.316	0.336	1
	1:00 AM	0.325	0.350	1
	2:00 AM	0.337	0.359	1
	3:00 AM	0.330	0.359	1
	4:00 AM	0.283	0.316	1
	5:00 AM	0.289	0.315	1
	6:00 AM	0.411	0.444	1
	7:00 AM	1.189	1.190	1
	8:00 AM	1.505	1.497	0
	9:00 AM	1.635	1.634	0
	10:00 AM	1.598	1.610	1
	11:00 AM	1.478	1.483	1
	12:00 PM	1.447	1.448	1
	1:00 PM	1.498	1.493	0
	2:00 PM	1.446	1.424	0
	3:00 PM	1.461	1.448	0
	4:00 PM	1.309	1.292	0
	5:00 PM	0.377	0.445	1
	6:00 PM	0.303	0.325	1
	7:00 PM	0.260	0.286	1
	8:00 PM	0.321	0.348	1
	9:00 PM	0.252	0.296	1
	10:00 PM	0.246	0.287	1
	11:00 PM	0.264	0.308	1

Table 6.16: Root mean cross-validation error (kWh) model comparison: 9200800

Location ID	Hour	Autoreg RMCVE	LOESS RMCVE	Model Indicator
9200800	12:00 AM	0.743	0.647	0
	1:00 AM	0.674	0.612	0
	2:00 AM	0.721	0.620	0
	3:00 AM	0.672	0.593	0
	4:00 AM	0.590	0.594	1
	5:00 AM	0.632	0.609	0
	6:00 AM	0.799	0.786	0
	7:00 AM	1.010	0.968	0
	8:00 AM	1.277	1.117	0
	9:00 AM	1.481	1.363	0
	10:00 AM	1.661	1.539	0
	11:00 AM	1.715	1.696	0
	12:00 PM	1.809	1.625	0
	1:00 PM	1.749	1.534	0
	2:00 PM	2.004	1.476	0
	3:00 PM	1.997	1.422	0
	4:00 PM	1.822	1.382	0
	5:00 PM	1.682	1.312	0
	6:00 PM	1.623	1.256	0
	7:00 PM	1.490	1.169	0
	8:00 PM	1.509	1.079	0
	9:00 PM	1.306	1.032	0
	10:00 PM	0.919	0.867	0
	11:00 PM	0.764	0.759	0

Table 6.17: Root mean cross-validation error (kWh) model comparison: 9201610

Location ID	Hour	Autoreg RMCVE	LOESS RMCVE	Model Indicator
9201610	12:00 AM	1.966	1.715	0
	1:00 AM	1.901	1.675	0
	2:00 AM	1.929	1.659	0
	3:00 AM	1.911	1.655	0
	4:00 AM	1.954	1.665	0
	5:00 AM	1.893	1.611	0
	6:00 AM	1.786	1.590	0
	7:00 AM	1.670	1.589	0
	8:00 AM	1.755	1.612	0
	9:00 AM	1.565	1.475	0
	10:00 AM	1.577	1.503	0
	11:00 AM	1.650	1.617	0
	12:00 PM	1.578	1.522	0
	1:00 PM	1.654	1.613	0
	2:00 PM	1.718	1.687	0
	3:00 PM	1.805	1.765	0
	4:00 PM	1.886	1.835	0
	5:00 PM	1.777	1.694	0
	6:00 PM	1.647	1.532	0
	7:00 PM	1.584	1.500	0
	8:00 PM	1.546	1.462	0
	9:00 PM	1.556	1.490	0
	10:00 PM	1.686	1.614	0
	11:00 PM	1.744	1.590	0

Table 6.18: Root mean cross-validation error (kWh) model comparison: 9202469

Location ID	Hour	Autoreg RMCVE	LOESS RMCVE	Model Indicator
9202469	12:00 AM	1.158	1.145	0
	1:00 AM	1.051	1.011	0
	2:00 AM	0.938	0.895	0
	3:00 AM	0.897	0.857	0
	4:00 AM	0.928	0.897	0
	5:00 AM	0.963	0.863	0
	6:00 AM	0.760	0.776	1
	7:00 AM	1.168	1.118	0
	8:00 AM	0.760	0.752	0
	9:00 AM	0.709	0.726	1
	10:00 AM	0.733	0.744	1
	11:00 AM	0.716	0.721	1
	12:00 PM	0.837	0.844	1
	1:00 PM	0.755	0.763	1
	2:00 PM	0.799	0.791	0
	3:00 PM	0.793	0.772	0
	4:00 PM	0.742	0.705	0
	5:00 PM	0.960	0.939	0
	6:00 PM	0.740	0.724	0
	7:00 PM	0.923	0.906	0
	8:00 PM	1.023	1.010	0
	9:00 PM	1.157	1.138	0
	10:00 PM	1.216	1.179	0
	11:00 PM	1.115	1.121	1

Table 6.19: Root mean cross-validation error (kWh) model comparison: 9202473

Location ID	Hour	Autoreg RMCVE	LOESS RMCVE	Model Indicator
9202473	12:00 AM	1.745	1.695	0
	1:00 AM	1.542	1.482	0
	2:00 AM	1.354	1.308	0
	3:00 AM	1.137	1.084	0
	4:00 AM	1.151	1.111	0
	5:00 AM	1.063	0.995	0
	6:00 AM	0.855	0.852	0
	7:00 AM	1.229	1.201	0
	8:00 AM	1.301	1.269	0
	9:00 AM	1.479	1.440	0
	10:00 AM	1.647	1.624	0
	11:00 AM	1.546	1.553	1
	12:00 PM	1.728	1.697	0
	1:00 PM	1.916	1.867	0
	2:00 PM	1.930	1.879	0
	3:00 PM	1.868	1.849	0
	4:00 PM	1.967	1.918	0
	5:00 PM	1.819	1.773	0
	6:00 PM	1.981	1.931	0
	7:00 PM	1.866	1.849	0
	8:00 PM	1.887	1.878	0
	9:00 PM	2.059	2.046	0
	10:00 PM	2.113	2.080	0
	11:00 PM	1.968	1.936	0

Table 6.20: Root mean cross-validation error (kWh) model comparison: 9202475

Location ID	Hour	Autoreg RMCVE	LOESS RMCVE	Model Indicator
9202475	12:00 AM	1.315	1.264	0
	1:00 AM	1.267	1.218	0
	2:00 AM	1.323	1.203	0
	3:00 AM	1.396	1.164	0
	4:00 AM	1.484	1.234	0
	5:00 AM	1.413	1.185	0
	6:00 AM	1.131	1.096	0
	7:00 AM	1.108	1.091	0
	8:00 AM	1.150	1.068	0
	9:00 AM	1.069	1.030	0
	10:00 AM	1.139	1.119	0
	11:00 AM	1.215	1.186	0
	12:00 PM	1.097	1.061	0
	1:00 PM	1.134	1.093	0
	2:00 PM	1.097	1.053	0
	3:00 PM	1.087	1.042	0
	4:00 PM	1.027	0.995	0
	5:00 PM	1.060	1.032	0
	6:00 PM	1.357	1.305	0
	7:00 PM	1.356	1.304	0
	8:00 PM	1.171	1.144	0
	9:00 PM	1.107	1.045	0
	10:00 PM	1.113	1.063	0
	11:00 PM	1.125	0.983	0

BIBLIOGRAPHY

[1] William S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *J. Amer. Statist. Assoc.*, 74(368):829–836, 1979.

[2] William S. Cleveland and Susan J. Devlin. Locally weighted regression: An approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 83(403):pp. 596–610, 1988.

[3] Vinod H.D. Generalization of the Durbin-Watson statistic for higher order autoregressive process. *Communication in Statistics*, 2:pp. 115–144, 1973.

[4] Clifford M. Hurvich, Jeffrey S. Simonoff, and Chih-Ling Tsai. Smoothing parameter selection in nonparametric regression using an improved akaike information criterion. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 60(2):pp. 271–293, 1998.

[5] M. C. Jones. Expressions for inverse moments of positive quadratic forms in normal variables. *Australian Journal of Statistics*, 28(2):242–250, 1986.

[6] M. C. Jones. On moments of ratios of quadratic forms in normal variables. *Statistics and Probability Letters*, 6(2):129–136, 1987.

[7] et al. Kutner M. *Applied Linear Statistical Models*. 5th Edition. McGraw-Hill/Irwin, New York, 2005.

[8] S. Massoud Amin and B.F. Wollenberg. Toward a smart grid: power delivery for the 21st century. *Power and Energy Magazine, IEEE*, 3(5):34 – 41, sept.-oct. 2005.

[9] W. Mendenhall and T. Sincich. *A Second Course in Statistics: Regression Analysis*. 6th Edition. Prentice Hall, Inc., New York, 2003.

[10] University of Kentucky. National weather service at university of kentucky (www.something.com), August 2011.

[11] SAS Institute Inc. *SAS/STAT Software, Version 9.2*. Cary, NC, 2008.